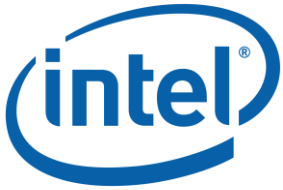


Intel® Core™2 Duo and Intel® Core™2 Solo Processor for Intel® Centrino® Duo Processor Technology

Intel® Celeron® Processor 500 Series

Specification Update

Including Intel® Core™2 Duo Extreme Edition Processors

March 2010

Revision 024

Doc. # 314079-024

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

64-bit computing on Intel architecture requires a computer system with a processor, chipset, BIOS, operating system, device drivers and applications enabled for Intel® 64 architecture. Processors will not operate (including 32-bit operation) without an Intel® 64 architecture-enabled BIOS. Performance will vary depending on your hardware and software configurations. Consult with your system vendor for more information.

+ System performance, battery life, high-definition quality and functionality, and wireless performance and functionality will vary depending on your specific operating system, hardware and software configurations. References to enhanced performance as measured by SySMark* 2004, PCMark* 2005 and 3DMark* 2005 refer to comparisons with previous generation Intel® Centrino® mobile technology platforms. References to improved battery life as measured by MobileMark* 2005, if applicable, refer to previous generation Intel Centrino mobile technology platforms. Wireless connectivity and some features may require you to purchase additional software, services or external hardware. Availability of public wireless LAN access points is limited, wireless functionality may vary by country and some hotspots may not support Linux-based Intel Centrino mobile technology systems. See www.intel.com/products/centrino/ for more information.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Intel, Intel Core2 Duo, Intel Core2 Solo, Intel Centrino Duo, Intel SpeedStep, MMX and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2006 – 2010, Intel Corporation. All rights reserved.

Contents

Preface 7
Identification Information 10
Summary Tables of Changes 19
Errata 42
Specification Changes 94
Specification Clarifications 96
Documentation Changes 97

Revision History

Document Number	Revision	Version	Description	Date
594274	-001	1.0	Initial release of Intel® Core™ 2 Duo Processor Preliminary Specification Update – NDA	July 2006
594274	-002	1.0	• Updated Sample Identification Table • Updated Errata AH19, AH20, AH29, AH38, AH40 and AH62 • Added Errata AH63 to AH75, AH5S	August 2006
594274	-003	1.0	• Updated Errata AH68, AH69, AH71, AH74 • Added Errata AH76 to AH82, AH6S • Added Specification Clarification on: —T75, T76 Specification —Maximum delta between Intel® Enhanced Deeper Sleep State and LFM on Intel® Core™2 Duo Mobile Processor for Intel® Centrino® Duo Mobile Technology production parts based on B-2 stepping	September 2006
594274	-004	1.0	• Updated Errata AH46, AH53 and AH78 • Added Errata AH83, AH84, AH85 and AH7S • Removed Erratum AH10 • Updated "Related Documents" Table Updated "Component Marking Information" Table	October 2006
594274	-005	1.0	• Updated Errata AH8, AH74, AH84, AH1S • Added Errata AH86, AH87, AH88, AH89, and AH90 • Removed Erratum AH63	November 2006
594274	-006	1.0	• Updated Erratum AH7S • Added Errata AH91-AH94 • Updated "Summary Tables of Changes"	December 2006
594274	-007	1.0	• Added Errata AH95-AH97 • Updated Erratum AH69 • Added Erratum AH8S	January 2007
594274	-008	1.0	• Added Errata AH98-AH100 • Added L-2 stepping processor SKUs, errata and MCU • Added Erratum AH9S	February 2007
594274	-009	1.0	• Corrected MCU for B-2/L-2 stepping. • Updated Erratum AH30, AH33 • Added Errata AH101-AH104 • Added new processor SKUs	March 2007
594274	-010	1.0	• Updated brand names	April 2007

Document Number	Revision	Version	Description	Date
			• Updated Summary Table of changes • Revised Errata AH14, AH25, AH26 • Added new Microcode Updates for B2 and L2 stepping • Added Errata AH105 • Added Specification Clarification AH3	
594274	-011	1.0	• Updated the following to include Intel Core 2 Duo processors on platforms based on the Mobile Intel 965 Express Chipset Family —Added Table 2, Intel® Core™ 2 Duo Processor for Mobile Intel® 965 Express Chipset Family Component Markings —Added Table 4, Intel® Core™2 Duo Processor for Mobile Intel® 965 Express Chipset Family-Microcode Update Guide —Added section, "Erratum for Intel® Core™ 2 Duo Processors for Platforms Based on Mobile Intel® 965 Express Chipset Family" • Removed Specification Clarification AH1 and AH2 • Removed AH3S. Fixed • Updated Specification Clarification AH3 • Updated Figure 1 and added Figure 2 • Added table linking processor stepping to CPU signature in the 'Component Identification Via the Programming Interface' • Added Errata AH106, AH107, AH5P, and AH10S	May 2007
594274	-012	1.0	• Updated Hyperlinks for the SDM Collateral under the "Related Documents" section • Added A-1 and G-0 Stepping to the Component Marking section • Added U2100 and U2200 A-1 stepping SKUs (QDF) to 945-based platform list. • Added G-0 and M-0 SV SKUs (QDF) to 965-based platform list • Added A-1 and G-0 stepping MCU • Added A-1 and G-0 stepping errata • Added new errata AH108-AH110, AH11S	June 2007
594274	-013	1.0	• Added Specification Change AP1 • Added G-0 L7500, L7700 processors • Added G-0 X7900 Extreme Edition processor • Added new errata AH111, AH6P	July 2007
594274	-014	1.0	• Added E-1 X7800 Extreme Edition Processor • Added E-1 X7900 Extreme Edition Processor • Added Production stepping G-O SV parts	August 2007
594274	-015	1.0	• Added AH112 – AH116	September 2007
355615	-016	1.0	• Added AH117 • Added SLV3V for Micro-FCBGA U7700	October 2007

Document Number	Revision	Version	Description	Date
355615	-017	1.0	• Added AH118 • Updated AH8 • Updated Summary Table of Changes	November 2007
355615	-018	1.0	• Added AH119	December 2007
355615	-019	1.0	• Added AH120 • Updated AH48 and AH51	January 2008
355615	-020	1.0	• Added AH121	July 2008
355615	-021	1.0	• Added SLGFJ, SLGFV for Micro-FCPGA T7400 • Added SLGFX for Micro-FCPGA L7400	September 2008
355615	-022	1.0	• Added Celeron Series processor information to the documentation	September 2008
355615	-023	1.0	• Added Errata for Intel® Celeron® Processor 500 series for Platforms based on the Intel® 965 Express Chipset Family (to include a new errata table). • Updated Specification Clarifications Table • Added G-2 L7500, T7400, T7500, L7400 Processors • Added M-1 U7500 Processor • Added M-1 573 Processor • Added M-1 and G-2 stepping MCU for Intel® Core™2 Duo Processors for Platforms based on the Intel® 945 Express Chipset Family • Added M-1 stepping MCU for Intel® Celeron® Processor 500 series for Platforms based on the Intel® 965 Express Chipset Family • Added M-0 U7700 Processor • Updated M-0 U7500, U7600 Processor	January 2010
355615	-024	1.0	• Added AH122	March 2010

§

Preface

This document is an update to the specifications contained in the documents listed in the following Affected Documents/Related Documents table. It is a compilation of device and document errata and specification clarifications and changes, and is intended for hardware system manufacturers and for software developers of applications, operating system, and tools.

Information types defined in the Nomenclature section of this document are consolidated into this update document and are no longer published in other documents. This document may also contain information that has not been previously published.

Affected Documents

Document Title	Document Number/Location
Intel® Core™ 2 Duo Processors and Intel® Core™ 2 Extreme Processors for Platforms Based on Mobile Intel® 965 Express Chipset Family Electrical, Mechanical, and Thermal Specification (EMTS)	Contact your Intel representative for the latest revision.
Intel® Core™2 Duo Mobile Processor for Intel® Centrino® Duo Technology Electrical, Mechanical, and Thermal Specification (EMTS)	Contact your Intel representative for the latest revision.
Intel® Celeron® Processor 500 Series for Platforms Based on Mobile Intel® 965 Express Chipset Family Datasheet	316205

Related Documents

Document Title (Doc number)	Location
Intel® 64 and IA-32 Architectures Software Developer's Manual Documentation Changes (Doc: 252046)	http://www.intel.com/design/processor/specupdt/252046.htm
Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 1: Basic Architecture (Doc: 253665)	http://www.intel.com/products/processor/manuals/index.htm
Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 2A: Instruction Set Reference, A-M (Doc: 253666)	http://www.intel.com/products/processor/manuals/index.htm
Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 2B: Instruction Set Reference, N-Z (Doc: 253667)	http://www.intel.com/products/processor/manuals/index.htm

Document Title (Doc number)	Location
Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3A: System Programming Guide (Doc: 253668)	http://www.intel.com/products/processor/manuals/index.htm
Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3B: System Programming Guide (Doc: 253669)	http://www.intel.com/products/processor/manuals/index.htm
IA-32 Intel® Architecture Optimization Reference Manual (Doc: 248966)	http://www.intel.com/products/processor/manuals/index.htm
Intel Processor Identification and the CPUID Instruction Application Note (AP-485) (Doc: 241618)	http://developer.intel.com/design/processor/applnots/241618.htm
Intel® 64 and IA-32 Architectures Application Note TLBs, Paging-Structure Caches, and Their Invalidation (Doc: 317080)	http://www.intel.com/design/processor/applnots/317080.pdf

Nomenclature

S-Spec Number is a five-digit code used to identify products. Products are differentiated by their unique characteristics (for example, core speed, L2 cache size, package type, etc.) as described in the processor identification information table. Care should be taken to read all notes associated with each S-Spec number.

QDF Number is a several digit code that is used to distinguish between engineering samples. These processors are used for qualification and early design validation. The functionality of these parts can range from mechanical only to fully functional. The NDA specification update has a processor identification information table that lists these QDF numbers and the corresponding product sample details.

Errata are design defects or errors. Errata may cause the processor's behavior to deviate from published specifications. Hardware and software designed to be used with any given stepping must assume that all errata documented for that stepping are present on all devices.

Specification Changes are modifications to the current published specifications. These changes will be incorporated in the next release of the specifications.

Specification Clarifications describe a specification in greater detail or further highlight a specification's impact to a complex design situation. These clarifications will be incorporated in the next release of the specifications.

Documentation Changes include typos, errors, or omissions from the current published specifications. These changes will be incorporated in the next release of the specifications.

Note: Errata remain in the specification update throughout the product's lifecycle, or until a particular stepping is no longer commercially available. Under these circumstances, errata removed from the specification update are archived and available upon request. Specification changes, specification clarifications and documentation changes

are removed from the specification update when the appropriate changes are made to the appropriate product specification or user documentation (datasheets, manuals, etc.).

§

Identification Information

Component Identification via Programming Interface

The Intel® Core™2 Duo mobile processor and Intel® Celeron® Processor 500 series stepping can be identified by the following register contents:

Family[1]	Model[2]	Model for A-1 step
0110	1111	10000

NOTES:
1. The family corresponds to bits [11:8] of the EDX register after RESET, bits [11:8] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register, and the generation field of the Device ID registers accessible through boundary scan.
2. The model corresponds to bits [7:4] of the EDX register after RESET, bits [7:4] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register, and the model field of the device ID registers accessible through boundary scan.

Cache and TLB descriptor parameters are provided in the EAX, EBX, ECX, and EDX registers after the CPUID instruction is executed with a 2 in the EAX register. Refer to the *Intel® Processor Identification and the CPUID Instruction Application Note* (AP-485) and the latest *Intel® Core™ 2 Duo Mobile Processor BIOS Writer's Guide* for further information on the CPUID instruction.

Each stepping of the Intel® Core™2 Duo mobile processor can be identified by software with its CPU signature:

Table 1. CPU Signature for the Intel® Core™2 Duo Mobile Processor

Stepping	CPU Signature
B-2	06F6h
L-2	06F2h
A-1	10661h
E-1	06FAh
M-1	06FDh
G-2	06FBh

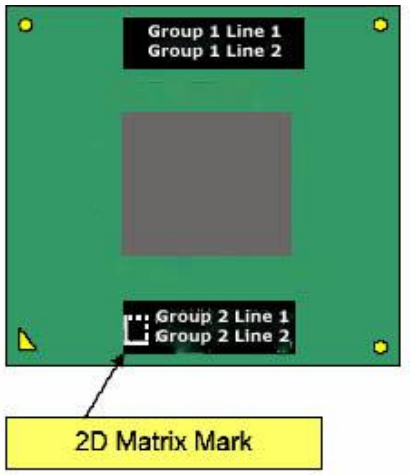
Each stepping of the Intel® Celeron® Processor 500 series can be identified by software with its CPU signature:

Table 2. CPU Signature for the Intel® Celeron® Processor

Stepping	CPU Signature
A-1	10661h
E-1	06FAh
M-1	06FDh

Component Marking Information

Figure 1. Intel® Core™2 Duo Mobile Processor (Micro-FCPGA/FCBGA) Sample Markings



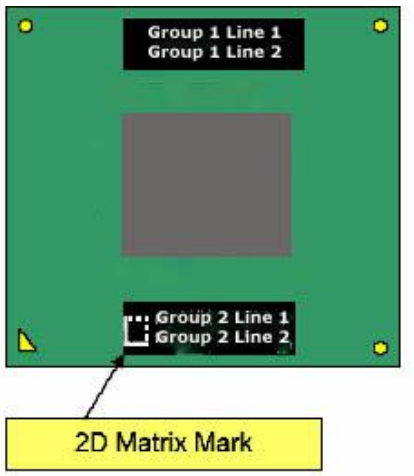
SAMPLE MARK EXAMPLE:
Group 1 Line 1: Unit Identifier* ---
Group 1 Line 2: FPO QDF# ES

Group 2 Line 1: INTEL (m) © ’05
Group 2 Line 2: ATPO Serial Number

For Pb-Free:
Group 2 Line 1: INTEL (m) © ’05 (e1)

* Intel Core 2 Duo mobile processors have a unit identifier of LF80537 for Micro-FCPGA parts and LE80537 for Micro-FCBGA parts.

Figure 2. Intel® Core™2 Duo Mobile Processor (Micro-FCPGA/FCBGA) Production Markings



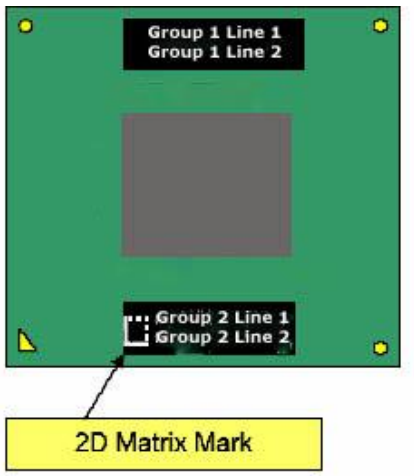
Production MARK EXAMPLE:
Group 1 Line 1: Unit Identifier* Processor #
Group 1 Line 2: FPO SSPEC#

Group 2 Line 1: Frequency/L2 Cache/FSB Speed
Group 2 Line 2: INTEL (m) © ’06

For Pb-Free:
Group 2 Line 2: INTEL (m) © ’06 (e1)

* Intel Core 2 Duo mobile processors have a unit identifier of LF80537 for Micro-FCPGA parts and LE80537 for Micro-FCBGA parts.

Figure 3. Intel® Celeron® Processor 500 Series (Micro-FCPGA) Markings



SAMPLE MARK EXAMPLE:
Group 1 Line 1: Unit Identifier ---
Group 1 Line 2: FPO S-Spec#

Group 2 Line 1: INTEL (m) © ’05
Group 2 Line 2: ATPO Serial Number

For Pb-Free:
Group 2 Line 1: INTEL (m) © ’05 (e1)

Table 3. Intel® Core™2 Duo Processor – Mobile Intel® 945 Express Chipset Family Component Markings

QDF#/ S-Spec	Processor Number	Package	Processor Stepping	FSB (MHz)	Speed HFM/LFM (GHz)	Notes
QTCY	n/a	Micro-FCBGA	B-2	667	1.50/1.00	3,5
QTDC	n/a	Micro-FCBGA	B-2	667	1.33/1.00	3,5
QTCA	n/a	Micro-FCPGA	B-2	667	2.33/1.00	1,3
QTDG	n/a	Micro-FCBGA	B-2	667	2.33/1.00	1,3
QTCE	n/a	Micro-FCPGA	B-2	667	2.17/1.00	1,3
QTDK	n/a	Micro-FCBGA	B-2	667	2.17/1.00	1,3
QTCI	n/a	Micro-FCPGA	B-2	667	2.00/1.00	1,3
QTDO	n/a	Micro-FCBGA	B-2	667	2.00/1.00	1,3
QTBS	n/a	Micro-FCPGA	B-2	667	1.83/1.00	2,3
QTCM	n/a	Micro-FCBGA	B-2	667	1.83/1.00	2,3
QTBW	n/a	Micro-FCPGA	B-2	667	1.66/1.00	2,3
QUOW	n/a	Micro-FCBGA	B-2	667	1.66/1.00	2,3
SL9SD	T7600	Micro-FCPGA	B-2	667	2.33/1.00	1,3
SL9SJ	T7600	Micro-FCBGA	B-2	667	2.33/1.00	1,3
S LGFJ	T7400	Micro-FCPGA	G-2	667	2.16/1.00	1,3
SL9SE	T7400	Micro-FCPGA	B-2	667	2.17/1.00	1,3
SLGFV	T7400	Micro-FCBGA	G-2	667	2.16/1.00	1,3
SL9SK	T7400	Micro-FCBGA	B-2	667	2.17/1.00	1,3
SL9SF	T7200	Micro-FCPGA	B-2	667	2.00/1.00	1,3
SL9SL	T7200	Micro-FCBGA	B-2	667	2.00/1.00	1,3
SL9SG	T5600	Micro-FCPGA	B-2	667	1.83/1.00	2,3
SL9SP	T5600	Micro-FCBGA	B-2	667	1.83/1.00	2,3
SL9SH	T5500	Micro-FCPGA	B-2	667	1.66/1.00	2,3
SL9SQ	T5500	Micro-FCBGA	B-2	667	1.66/1.00	2,3
QWMA	U7600	Micro-FCBGA	L-2	533	1.20/0.80	3,5
QUGS	U7500	Micro-FCBGA	L-2	533	1.06/0.80	3,5
SLAUT	U7500	Micro-FCBGA	M-0	533	1.06/0.80	3,5
SLV3X	U7500	Micro-FCBGA	M-1	533	1.06/0.80	3,5
QUFU	T5600	Micro-FCPGA	L-2	667	1.83/1.00	2,3
QUGK	T5600	Micro-FCBGA	L-2	667	1.83/1.00	2,3
QUGC	T5500	Micro-FCPGA	L-2	667	1.66/1.00	2,3

QDF#/ S-Spec	Processor Number	Package	Processor Stepping	FSB (MHz)	Speed HFM/LFM (GHz)	Notes
QUGO	T5500	Micro-FCBGA	L-2	667	1.66/1.00	2,3
SLV3W	U7600	Micro-FCBGA	M-0	533	1.20/0.80	3,5
SLAUS	U7600	Micro-FCBGA	M-0	533	1.20/0.80	3,5
SL9U3	T5600	Micro-FCPGA	L-2	667	1.83/1.00	2,3
SL9U7	T5600	Micro-FCBGA	L-2	667	1.83/1.00	2,3
SL9U4	T5500	Micro-FCPGA	L-2	667	1.66/1.00	2,3
SL9U8	T5500	Micro-FCBGA	L-2	667	1.66/1.00	2,3
SL9WE	T5300	Micro-FCPGA	L-2	533	1.73/0.80	2,3
QUFY	n/a	Micro-FCPGA	L-2	533	1.73/0.80	2,3
QTRJ	n/a	Micro-FCPGA	B-2	533	1.60/0.80	2,3
SL9VP	T5200	Micro-FCPGA	B-2	533	1.60/0.80	2,3
S LGFX	L7400	Micro-FCBGA	G-2	667	1.50/1.00	3,4
SL9SM	L7400	Micro-FCBGA	B-2	667	1.50/1.00	3,4
SL9SN	L7200	Micro-FCBGA	B-2	667	1.33/1.00	3,4
QZJC	U2100	Micro-FCBGA	A-1	533	1.06/0.80	3,6
QZIX	U2200	Micro-FCBGA	A-1	533	1.20/0.80	3,6
SLAUR	U7700	Micro-FCBGA	M-0	533	1.33/1.00	3,5

NOTES:

1. Intel® Core™2 Duo processor Standard Voltage with 4-M L2 cache.
2. Intel® Core™2 Duo processor Standard Voltage with 2-M L2 cache.
3. V_{CC_CORE} = (1.250-1.075)/ (0.95-0.85) V for Highest/Lowest (HFM/LFM) / Deeper Sleep VID = (0.75-0.65) V / Intel® Enhanced Deeper Sleep VID = (0.65) V.
4. Intel® Core™2 Duo processor Low Voltage with 4-M L2 cache.
5. Intel® Core™2 Duo processor Ultra Low Voltage with 2-M L2 cache.
6. Intel® Core™2 Solo processor Ultra Low Voltage with 1-M L2 cache.

Table 2. Intel® Core™2 Duo Processor – Mobile Intel® 965 Express Chipset Family Component Markings

QDF#/ S-Spec	Processor Number	Package	Processor Stepping	FSB (MHz)	Speed IDA[3]/HFM/LFM/SLFM[4] (GHz)	Notes
QXJJ	T7700	Micro-FCPGA	E-1	800	2.60/**2.40**/1.20/0.80	1,2,3,6
QXJO	T7700	Micro-FCBGA	E-1	800	2.60/**2.40**/1.20/0.80	1,2,3,6
QXJK	T7500	Micro-FCPGA	E-1	800	2.40/**2.20**/1.20/0.80	1,2,3,5
QXJP	T7500	Micro-FCBGA	E-1	800	2.40/**2.20**/1.20/0.80	1,2,3,5

QDF#/ S-Spec	Processor Number	Package	Processor Stepping	FSB (MHz)	Speed IDA[3]/HFM/LFM/SLFM[4] (GHz)	Notes
QXJL	T7300	Micro-FCPGA	E-1	800	2.20/**2.00**/1.20/0.80	1,2,3,4
QXJQ	T7300	Micro-FCBGA	E-1	800	2.20/**2.00**/1.20/0.80	1,2,3,4
QVXK	L7500	Micro-FCBGA	E-1	800	1.80/**1.60**/1.20/0.80	2,3,7,8
QVXO	L7300	Micro-FCBGA	E-1	800	1.60/**1.40**/1.20/0.80	2,3,7,8
QWOY	T7100	Micro-FCPGA	M-0	800	2.00/**1.80**/1.20/0.80	2,3,6,9
QWOI	T7100	Micro-FCBGA	M-0	800	2.00/**1.80**/1.20/0.80	2,3,6,9
QWQE	U7600	Micro-FCBGA	M-0	533	1.33/**1.20**/0.80/NA	2,10,11
QWQI	U7500	Micro-FCBGA	M-0	533	1.20/**1.06**/0.80/NA	2,10,11
SLA43	T7700	Micro-FCPGA	E-1	800	2.60/**2.40**/1.20/0.80	1,2,3,6
SLA3M	T7700	Micro-FCBGA	E-1	800	2.60/**2.40**/1.20/0.80	1,2,3,6
SLA44	T7500	Micro-FCPGA	E-1	800	2.40/**2.20**/1.20/0.80	1,2,3,5
SLA3N	T7500	Micro-FCBGA	E-1	800	2.40/**2.20**/1.20/0.80	1,2,3,5
SLA45	T7300	Micro-FCPGA	E-1	800	2.20/**2.00**/1.20/0.80	1,2,3,4
SLA3P	T7300	Micro-FCBGA	E-1	800	2.20/**2.00**/1.20/0.80	1,2,3,4
SLA4A	T7100	Micro-FCPGA	M-0	800	2.00/**1.80**/1.20/0.80	2,3,6,9
SLA3U	T7100	Micro-FCBGA	M-0	800	2.00/**1.80**/1.20/0.80	2,3,6,9
SLA3R	L7500	Micro-FCBGA	E-1	800	1.80/**1.60**/1.20/0.80	2,3,7,8
QXMV	L7500	Micro-FCBGA	G-0	800	1.80/**1.60**/1.20/0.80	2,3,7,8
QXMR	L7700	Micro-FCBGA	G-0	800	2.00/**1.80**/1.20/0.80	2,3,7,8
SLA3S	L7300	Micro-FCBGA	E-1	800	1.60/**1.40**/1.20/0.80	2,3,7,8

QDF#/ S-Spec	Processor Number	Package	Processor Stepping	FSB (MHz)	Speed IDA[3]/HFM/LFM/SLFM[4] (GHz)	Notes
SLV3W	U7600	Micro-FCBGA	M-0	533	1.33/**1.20**/0.80/NA	2,10,11
SLV3V	U7700	Micro-FCBGA	M-0	533	1.47/**1.33**/0.80/NA	2,10,11
SLAUT	U7500	Micro-FCBGA	M-1	533	1.20/**1.06**/0.80/NA	2,10,11
QXNH	T7800	Micro-FCPGA	G-0	800	2.8/**2.6**/1.2/0.8	1,2,3,6
QXLK	T7800	Micro-FCBGA	G-0	800	2.8/**2.6**/1.2/0.8	1,2,3,6
QXNM	T7700	Micro-FCPGA	G-0	800	2.6/**2.4**/1.2/0.8	1,2,3,6
QXLP	T7700	Micro-FCBGA	G-0	800	2.6/**2.4**/1.2/0.8	1,2,3,6
QXNR	T7500	Micro-FCPGA	G-0	800	2.4/**2.2**/1.2/0.8	1,2,3,6
QXLU	T7500	Micro-FCBGA	G-0	800	2.4/**2.2**/1.2/0.8	1,2,3,6
SLAF8	T7500	Micro-FCPGA	G-2	800	2.4/**2.2**/1.2/0.8	1,2,3,6
SLADM	T7500	Micro-FCBGA	G-2	800	2.4/**2.2**/1.2/0.8	1,2,3,6
QWOU	T7250	Micro-FCPGA	M-0	800	2.2/**2.0**/1.2/0.8	2,3,6,9
QWOE	T7250	Micro-FCBGA	M-0	800	2.2/**2.0**/1.2/0.8	2,3,6,9
QZDX	X7900	Micro-FCPGA	G-0	800	**2.8**/1.2/0.8	1,3
QVVF	X7800	Micro-FCPGA	E-1	800	**2.6**/1.2/0.8	1,3
SLAF4	X7900	Micro-FCPGA	G-0	800	**2.8**/1.2/0.8	1,3
SLA33	X7900	Micro-FCPGA	E-1	800	**2.8**/1.2/0.8	1,3
SLA6Z	X7800	Micro-FCPGA	E-1	800	**2.6**/1.2/0.8	1,3
SLAF6	T7800	Micro-FCPGA	G-0	800	2.8/**2.6**/1.2/0.8	1,2,3,6
SLA75	T7800	Micro-FCBGA	G-0	800	2.8/**2.6**/1.2/0.8	1,2,3,6
SLAF7	T7700	Micro-FCPGA	G-0	800	2.6/**2.4**/1.2/0.8	1,2,3,6
SLADL	T7700	Micro-FCBGA	G-0	800	2.6/**2.4**/1.2/0.8	1,2,3,6

QDF#/ S-Spec	Processor Number	Package	Processor Stepping	FSB (MHz)	Speed IDA[3]/HFM/LFM/SLFM[4] (GHz)	Notes
SLA49	T7250	Micro-FCPGA	M-0	800	2.2/**2.0**/1.2/0.8	2,3,6,9
SLA3T	T7250	Micro-FCBGA	M-0	800	2.2/**2.0**/1.2/0.8	2,3,6,9
SLAET	L7500	Micro-FCBGA	G-2	800	1.80/**1.60**/1.20/0.80	2,3,7,8
The Intel® Core™2 Duo Processor for Mobile Intel® 965 Express Chipset Family introduces several new features. The table has been changed to reflect the addition of features such as Intel® Dynamic Acceleration and Intel® Dynamic Front Side Bus Frequency Switching.						

NOTES:

1. Intel® Core™2 Duo mobile processor Standard Voltage with 4-M L2 cache.
2. Intel® Dynamic Acceleration (IDA) is supported. While one core is inactive, this feature allows one of the processor cores to temporarily operate at a higher frequency point than HFM when the operating system requests increased performance. This higher frequency is called the opportunistic frequency and the HFM frequency is called the guaranteed frequency. The opportunistic frequency has a bus ratio of (N+1):1, where N is the guaranteed frequency bus ratio.
3. Intel® Dynamic Front Side Bus Frequency Switching is supported. The P-state enabled by this feature, known as SuperLFM, has a frequency of 800 MHz and takes the place of legacy LFM as the lowest voltage and frequency point. This feature requires BIOS enabling in both the CPU and a Mobile Intel 965 Express Chipset Family part. Please contact an Intel representative for more details.
4. V_{CC_CORE} = [1.3125-1.0750] V for IDA mode, [1.250-1.075] / [1.0000-0.9375] V for Highest/Lowest Frequency Mode (HFM/LFM), [0.9-0.8] V for Super LFM, Deeper Sleep VID = [0.75-0.65] V, Intel Enhanced Deeper Sleep VID = [0.65-0.60] V.
5. V_{CC_CORE} = [1.2500-1.0750] V for IDA mode, [1.250-1.075] / [1.0000-0.9375] V for Highest/Lowest Frequency Mode (HFM/LFM), [0.9-0.8] V for Super LFM, Deeper Sleep VID = [0.75-0.65] V, Intel Enhanced Deeper Sleep VID = [0.65-0.60] V.
6. V_{CC_CORE} = [1.2500-1.0750] V for IDA mode, [1.175-1.075] / [1.0000-0.9375] V for Highest/Lowest Frequency Mode (HFM/LFM), [0.9-0.8] V for Super LFM, Deeper Sleep VID = [0.75-0.65] V, Intel Enhanced Deeper Sleep VID = [0.65-0.60] V.
7. Intel Core 2 Duo mobile processor Low Voltage with 4-M L2 cache.
8. V_{CC_CORE} = [1.1500-0.9750] V for IDA mode, [1.0625-0.9750] / [0.9750] V for Highest/Lowest Frequency Mode (HFM/LFM), [0.90-0.85] V for Super LFM, Deeper Sleep VID = [0.75-0.65] V, Intel Enhanced Deeper Sleep VID = [0.65] V.
9. Intel Core 2 Duo mobile processor Standard Voltage with 2-M L2 Cache.
10. Intel Core 2 Duo mobile processor Ultra Low Voltage with 2-M L2 Cache.
11. V_{CC_CORE} = [1.1000-0.9750] V for IDA mode, [0.9750-0.8500] / [0.8500] V for Highest/Lowest Frequency Mode (HFM/LFM), Deeper Sleep VID = [0.75-0.65] V, Intel Enhanced Deeper Sleep VID = [0.65-0.60] V

Table 1. Intel® Celeron® Processor 500 Series Component Markings

QDF#/ S-Spec	Package	Processor Number	Processor Stepping	FSB (MHz)	Speed (GHz)	Voltage (V)	Notes
QVTG	Micro-FCPGA	n/a	A-1	533	1.73	1.30–0.95	1,5
QVTB	Micro-FCPGA	n/a	A-1	533	1.86	1.30–0.95	1,5

QDF#/ S-Spec	Package	Processor Number	Processor Stepping	FSB (MHz)	Speed (GHz)	Voltage (V)	Notes
QVTH	Micro-FCPGA	530	A-1	533	1.73	1.30-0.95	1,5
QZVO	Micro-FCPGA	530	A-1	533	1.73	1.30-0.95	1,4
SLA2G	Micro-FCPGA	530	A-1	533	1.73	1.30-0.95	1,5
SL9VA	Micro-FCPGA	530	A-1	533	1.73	1.30-0.95	1,4
SLA48	Micro-FCPGA	530	E-1	533	1.73	1.30-0.95	2,5
QWLO	Micro-FCPGA	530	E-1	533	1.73	1.250-0.975	2,5
SLGFL	Micro-FCPGA	530	G-2	533	1.73	1.250-0.975	1,4
SLGFY	Micro-FCBGA	530	G-2	533	1.73	1.250-0.975	1,4
SLA2F	Micro-FCPGA	540	A-1	533	1.86	1.30–0.95	1,5
SLA47	Micro-FCPGA	540	E-1	533	1.86	1.30–0.95	2,5
SLAJ9	Micro-FCPGA	550	G-2	533	2.00	1.30–0.95	1,5
SLALD	Micro-FCBGA	550	G-2	533	2.00	1.30-0.95	1,5
SLA2E	Micro-FCPGA	550	A-1	533	2.00	1.30–0.95	1,5
QVTC	Micro-FCPGA	540	A-1	533	1.86	1.30–0.95	1,5
QXWY	Micro-FCPGA	540	E-1	533	1.86	1.250-0.975	2,5
QVSS	Micro-FCPGA	550	A-1	533	2.00	1.30-0.95	1,5
QVSN	Micro-FCPGA	560	A-1	533	2.13	1.30-0.95	1,5

NOTES:

1. Intel® Celeron® Processor 500 series Standard Voltage based on a single core, 1-M L2 cache.
2. Intel® Celeron® Processor 500 series Standard Voltage based on a fused single core, 1-M L2 cache.
3. Intel® Celeron® Processor 500 series Ultra Low Voltage based on a single core, 1-M L2 cache.
4. Socket-M processor for Intel® Centrino® Duo processor technology (Napa Refresh Platform).
5. Socket-P processor for Intel® Centrino® Duo processor technology (Santa Rosa Platform).

Summary Tables of Changes

The following table indicates the Specification Changes, Errata, Specification Clarifications or Documentation Changes, which apply to the listed CPU steppings. Intel intends to fix some of the errata in a future stepping of the component, and to account for the other outstanding issues through documentation or Specification Changes as noted. This table uses the following notations:

Codes Used in Summary Table

Stepping

X:	Erratum, Specification Change or Clarification that applies to this stepping.
(No mark) or (Blank Box):	This erratum is fixed in listed stepping or specification change does not apply to listed stepping.

Status

Doc:	Document change or update that will be implemented.
Plan Fix:	This erratum may be fixed in a future stepping of the product.
Fixed:	This erratum has been previously fixed.
No Fix:	There are no plans to fix this erratum.

Row

Shaded:	This item is either new or modified from the previous version of the document.

Note: Each Specification Update item is prefixed with a capital letter to distinguish the product. The key below details the letters that are used in Intel's microprocessor Specification Updates:

A = Dual-Core Intel® Xeon® processor 7000$^{\Delta}$ sequence

C = Intel® Celeron® processor

D = Dual-Core Intel® Xeon® processor 2.80 GHz

E = Intel® Pentium® III processor

F = Intel® Pentium® processor Extreme Edition and Intel® Pentium® D processor

I = Dual-Core Intel® Xeon® processor 5000$^{\Delta}$ series

J = 64-bit Intel® Xeon® processor MP with 1-MB L2 cache

K = Mobile Intel® Pentium® III processor

L = Intel® Celeron® D processor

M = Mobile Intel® Celeron® processor

N = Intel ® Pentium® 4 processor

O = Intel ® Xeon® processor MP

P = Intel ® Xeon® processor

Q = Mobile Intel® Pentium® 4 processor supporting Hyper-Threading Technology on 90-nm process technology

R = Intel® Pentium® 4 processor on 90-nm process

S = 64-bit Intel® Xeon® processor with 800-MHz system bus (1-MB and 2-MB L2 cache versions)

T = Mobile Intel® Pentium® 4 processor–M

U = 64-bit Intel® Xeon® processor MP with up to 8-MB L3 cache

V = Mobile Intel® Celeron® processor on .13 Micron Process in Micro-FCPGA Package

W= Intel® Celeron®-M processor

X = Intel® Pentium® M processor on 90-nm process with 2-MB L2 cache and Intel® Processors A100 and A110 with 512-kB L2 cache

Y = Intel® Pentium® M processor

Z = Mobile Intel® Pentium® 4 processor with 533-MHz system bus

AA= Intel® Pentium® D Processor 900$^{\Delta}$ Sequence and Intel® Pentium® processor Extreme Edition 955$^{\Delta}$, 965$^{\Delta}$

AB= Intel® Pentium® 4 processor 6x1 Sequence

AC= Intel® Celeron® processor in 478-pin package

AD = Intel® Celeron® D processor on 65-nm process

AE = Intel® Core™ Duo processor and Intel® Core™ Solo processor on 65-nm process

AF = Dual-Core Intel® Xeon® processor LV

AG = Dual-Core Intel® Xeon® processor 5100$^{\Delta}$ series

AH= Intel® Core™2 Duo/Solo processor for Intel® Centrino® Duo processor technology

AI = Intel® Core™2 Extreme processor X6800$^{\Delta}$ and Intel® Core™2 Duo Desktop processor E6000$^{\Delta}$ and E4000$^{\Delta}$ sequence

AJ = Quad-Core Intel® Xeon® processor 5300$^{\Delta}$ series

AK = Intel® Core™2 Extreme quad-core processor QX6000 sequence and Intel® Core™2 Quad processor Q6000 sequence

AL = Dual-Core Intel® Xeon® processor 7100 $^{\Delta}$ series

AM = Intel® Celeron® processor 400 sequence

AN = Intel® Pentium® Dual-Core processor

AO = Quad-Core Intel® Xeon® processor 3200Δ series

AP = Dual-Core Intel® Xeon® processor 3000Δ series

AQ = Intel® Pentium® Dual-Core Desktop processor E2000Δ sequence

AR = Intel® Celeron® Processor 500Δ series

AS = Intel® Xeon® processor 7200, 7300 series

AT = Intel® Celeron® processor 200 series

AV = Intel® Core™2 Extreme Processor QX9000 Sequence and Intel® Core™2 Quad Processor Q9000 Sequence processor

AX = Quad-Core Intel® Xeon® Processor 5400 Series

AY = Wolfdale DP

Δ Intel® processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. See http://www.intel.com/products/processor_number for details.

Errata for Intel® Core™2 Duo Processors for Platforms Based on Mobile Intel 945 Express Chipset Family

Number	Stepping	Stepping	Stepping	Plans	ERRATA
	B-2	L-2	A-1		
AH1	X	X	X	No Fix	Writing the Local Vector Table (LVT) When an Interrupt Is Pending May Cause an Unexpected Interrupt
AH2	X	X	X	No Fix	LOCK# Asserted During a Special Cycle Shutdown Transaction May Unexpectedly Deassert
AH3	X	X	X	No Fix	Address Reported by Machine-Check Architecture (MCA) on Single-bit L2 ECC Errors May Be Incorrect
AH4	X	X	X	No Fix	VERW/VERR/LSL/LAR Instructions May Unexpectedly Update the Last Exception Record (LER) MSR
AH5	X	X	X	No Fix	DR3 Address Match on MOVD/MOVQ/MOVNTQ Memory Store Instruction May Incorrectly Increment Performance Monitoring Count for Saturating SIMD Instructions Retired (Event CFH)
AH6	X	X	X	Fixed	SYSRET May Incorrectly Clear RF (Resume Flag) in the RFLAGS Register
AH7	X	X	X	No Fix	General Protection Fault (#GP) for Instructions Greater than 15 Bytes May Be Preempted
AH8	X	X	X	No Fix	Pending x87 FPU exceptions (#MF) following STI may be serviced before higher priority interrupts.
AH9	X	X	X	No Fix	The Processor May Report a #TS Instead of a #GP Fault
AH10					Removed Erratum
AH11	X	X	X	No Fix	A Write to an APIC Register Sometimes May Appear to Have Not Occurred
AH12	X	X	X	No Fix	Programming the Digital Thermal Sensor (DTS) Threshold May Cause Unexpected Thermal Interrupts
AH13	X	X	X	No Fix	Count Value for Performance-Monitoring Counter PMH_PAGE_WALK May Be Incorrect
AH14	X	X	X	No Fix	LER MSRs May be Incorrectly Updated
AH15	X	X	X	No Fix	Performance Monitoring Events for Retired Instructions (C0H) May Not Be Accurate
AH16	X	X	X	No Fix	Performance Monitoring Event For Number Of Reference Cycles When The Processor Is Not Halted (3CH) Does Not Count According To The Specification
AH17	X	X		Fixed	Using 2M/4M Pages When A20M# Is Asserted May Result in Incorrect Address Translations
AH18	X	X	X	No Fix	Writing Shared Unaligned Data that Crosses a Cache Line without Proper Semaphores or Barriers May Expose a Memory Ordering Issue

Number	Stepping	Stepping	Stepping	Plans	ERRATA
	B-2	L-2	A-1		
AH19	X	X	X	No Fix	Code Segment Limit Violation May Occur On 4 Gigabyte Limit Check
AH20	X	X	X	Plan Fix	FP Inexact-Result Exception Flag May Not Be Set
AH21	X	X	X	Plan Fix	Global Pages in the Data Translation Look-Aside Buffer (DTLB) May Not Be Flushed by RSM instruction before Restoring the Architectural State from SMRAM
AH22	X	X	X	Plan Fix	Sequential Code Fetch to Non-canonical Address May have Nondeterministic Results
AH23	X	X		Fixed	VMCALL to Activate Dual-monitor Treatment of SMIs and SMM Ignores Reserved Bit settings in VM-exit Control Field
AH24	X	X	X	No Fix	REP MOVS/STOS Executing with Fast Strings Enabled and Crossing Page Boundaries with Inconsistent Memory Types May Use an Incorrect Data Size or Lead to Memory-Ordering Violations.
AH25	X	X	X	No Fix	Some Bus Performance Monitoring Events May Not Count Local Events under Certain Conditions
AH26	X	X	X	No Fix	Premature Execution of a Load Operation Prior to Exception Handler Invocation
AH27	X	X	X	No Fix	General Protection (#GP) Fault May Not Be Signaled on Data Segment Limit Violation above 4-G Limit
AH28	X	X	X	No Fix	EIP May Be Incorrect after Shutdown in IA-32e Mode
AH29	X	X	X	No Fix	#GP Fault Is Not Generated on Writing IA32_MISC_ENABLE [34] When Execute Disable Is Not supported
AH30	X		X	Plan Fix	(E)CX May Get Incorrectly Updated Fast String REP MOVS or Fast String REP STOS with Large Data Structures
AH31	X	X	X	Plan Fix	Performance Monitoring Events for Retired Loads (CBH) and Instructions Retired (C0H) May Not Be Accurate
AH32	X	X	X	No Fix	Upper 32 bits of 'From' Address Reported through BTMs or BTSs May Be Incorrect
AH33	X	X	X	Plan Fix	Unsynchronized Cross-Modifying Code Operations Can Cause unexpected Instruction Execution Results
AH34	X	X	X	No Fix	MSRs Actual Frequency Clock Count (IA32_APERF) or Maximum Frequency Clock Count (IA32_MPERF) May Contain Incorrect Data after a Machine Check Exception (MCE)
AH35	X	X	X	No Fix	Incorrect Address Computed for Last Byte of FXSAVE/FXRSTOR Image Leads to Partial Memory Update
AH36	X	X	X	No Fix	Split Locked Stores May Not Trigger the Monitoring Hardware
AH37	X		X	Fixed	REP CMPS/SCAS Operations May Terminate Early in 64-bit Mode When RCX >= 0X100000000

Number	Stepping	Stepping	Stepping	Plans	ERRATA
	B-2	L-2	A-1		
AH38	X	X	X	Plan Fix	FXSAVE/FXRSTOR Instructions which Store to the End of the Segment and Cause a Wrap to a Misaligned Base Address (Alignment <= 0x10h) May Cause FPU Instruction or Operand Pointer Corruption
AH39	X	X		Fixed	Cache Data Access Request from One Core Hitting a Modified Line in the L1 Data Cache of the Other Core May Cause Unpredictable System Behavior
AH40	X	X	X	Plan Fix	PREFETCHh Instruction Execution under Some Conditions May Lead to Processor Livelock
AH41	X	X	X	Plan Fix	PREFETCHh Instructions May Not Be Executed when Alignment Check (AC) Is Enabled
AH42	X	X	X	Plan Fix	Upper 32 Bits of the FPU Data (Operand) Pointer in the FXSAVE Memory Image May Be Unexpectedly All 1's after FXSAVE
AH43	X	X		Fixed	Concurrent Multi-processor Writes to Non-dirty Page May Result in Unpredictable Behavior
AH44	X	X	X	Plan Fix	Performance Monitor IDLE_DURING_DIV (18h) Count May Not Be Accurate
AH45	X	X	X	No Fix	Values for LBR/BTS/BTM Will Be Incorrect after an Exit from SMM
AH46	X	X		Fixed	Shutdown Condition May Disable Non-Bootstrap Processors
AH47	X		X	Fixed	SYSCALL Immediately after Changing EFLAGS.TF May Not Behave According to the New EFLAGS.TF
AH48	X	X	X	No Fix	Code Segment Limit/Canonical Faults on RSM May be Serviced before Higher Priority Interrupts/Exceptions and May Push the Wrong Address Onto the Stack
AH49	X	X	X	No Fix	VM Bit Is Cleared on Second Fault Handled by Task Switch from Virtual-8086 (VM86)
AH50	X	X	X	Plan Fix	IA32_FMASK Is Reset during an INIT
AH51	X	X	X	No Fix	An Enabled Debug Breakpoint or Single Step Trap May Be Taken after MOV SS/POP SS Instruction if it is Followed by an Instruction That Signals a Floating Point Exception
AH52	X	X	X	No Fix	Last Branch Records (LBR) Updates May Be Incorrect after a Task Switch
AH53	X	X	X	No Fix	IO_SMI Indication in SMRAM State Save Area May Be Set Incorrectly
AH54	X	X	X	No Fix	INIT Does Not Clear Global Entries in the TLB
AH55	X	X	X	Plan Fix	Using Memory Type Aliasing with Memory Types WB/WT May Lead to Unpredictable Behavior
AH56	X	X	X	Plan Fix	Update of Read/Write (R/W) or User/Supervisor (U/S) or Present (P) Bits without TLB Shootdown May Cause Unexpected Processor Behavior

Number	Stepping	Stepping	Stepping	Plans	ERRATA
	B-2	L-2	A-1		
AH57	X	X	X	No Fix	BTS Message May Be Lost When the STPCLK# Signal Is Active
AH58	X	X	X	No Fix	MOV To/From Debug Registers Causes Debug Exception
AH59	X	X	X	No Fix	EFLAGS Discrepancy on a Page Fault after a Multiprocessor TLB Shootdown
AH60	X	X	X	No Fix	LBR, BTS, BTM May Report a Wrong Address when an Exception/Interrupt Occurs in 64-bit Mode
AH61	X	X	X	No Fix	A Thermal Interrupt Is Not Generated when the Current Temperature Is Invalid
AH62	X	X	X	No Fix	CMPSB, LODSB, or SCASB in 64-bit Mode with Count Greater or Equal to 2^{48} May Terminate Early
AH63					Removed Erratum
AH64	X	X	X	No Fix	Returning to Real Mode from SMM with EFLAGS.VM Set May Result in Unpredictable System Behavior
AH65	X	X		Plan Fix	VMLAUNCH/VMRESUME May Not Fail When VMCS Is Programmed to Cause VM Exit to Return to a Different Mode
AH66	X	X	X	No Fix	IRET under Certain Conditions May Cause an Unexpected Alignment Check Exception
AH67	X	X	X	No Fix	Performance Monitoring Event FP_ASSIST May Not Be Accurate
AH68	X	X	X	Plan Fix	CPL-Qualified BTS May Report Incorrect Branch-From Instruction Address
AH69	X	X	X	Plan Fix	PEBS Does Not Always Differentiate Between CPL-Qualified Events
AH70	X	X	X	No Fix	PMI May Be Delayed to Next PEBS Event
AH71	X	X	X	Plan Fix	PEBS Buffer Overflow Status Will Not Be Indicated Unless IA32_DEBUGCTL[12] Is Set
AH72	X	X	X	No Fix	The BS Flag in DR6 May Be Set for Non-Single-Step #DB Exception
AH73	X	X	X	No Fix	An Asynchronous MCE during a Far Transfer May Corrupt ESP
AH74	X	X		Plan Fix	In Single-Stepping on Branches Mode, the BS Bit in the Pending-Debug-Exceptions Field of the Guest State Area will be Incorrectly Set by VM Exit on a MOV to CR8 Instruction
AH75	X	X	X	No Fix	B0-B3 Bits in DR6 May Not Be Properly Cleared after Code Breakpoint
AH76	X	X	X	No Fix	BTM/BTS Branch-From Instruction Address May Be Incorrect for Software Interrupts
AH77	X	X	X	Plan Fix	REP Store Instructions in a Specific Situation May Cause the Processor to Hang

Number	Stepping	Stepping	Stepping	Plans	ERRATA
	B-2	L-2	A-1		
AH78	X	X	X	No Fix	Performance Monitor SSE Retired Instructions May Return Incorrect Values
AH79	X	X	X	No Fix	Performance Monitoring Events for L1 and L2 Miss May Not Be Accurate
AH80	X	X	X	No Fix	Store to WT Memory Data May Be Seen in Wrong Order by Two Subsequent Loads
AH81	X	X	X	No Fix	A MOV Instruction from CR8 Register with 16 Bit Operand Size Will Leave Bits 63:16 of the Destination Register Unmodified
AH82	X	X	X	Plan Fix	Debug Register May Contain Incorrect Information on a MOVSS or POPSS Instruction followed by SYSRET
AH83	X	X	X	No Fix	Single Step Interrupts with Floating Point Exception Pending May Be Mishandled
AH84	X	X		No Fix	Non-Temporal Data Store May Be Observed in Wrong Program Order
AH85	X	X	X	No Fix	Fault on ENTER Instruction May Result in Unexpected Values on Stack Frame
AH86	X	X		Plan Fix	CPUID Reports Architectural Performance Monitoring Version 2 is Supported, When Only Version 1 Capabilities are Available
AH87	X	X	X	No Fix	Unaligned Accesses to Paging Structures May Cause the Processor to Hang
AH88	X	X		No Fix	Microcode Updates Performed During VMX Non-root Operation Could Result in Unexpected Behavior
AH89	X	X	X	No Fix	INVLPG Operation for Large (2M/4M) Pages May be Incomplete under Certain Conditions
AH90	X	X	X	No Fix	Page Access Bit May be Set Prior to Signaling a Code Segment Limit Fault
AH91	X	X	X	Plan Fix	Update of Attribute Bits on Page Directories without Immediate TLB Shootdown May Cause Unexpected Processor Behavior
AH92	X	X	X	Plan Fix	Invalid Instructions May Lead to Unexpected Behavior
AH93	X	X	X	No Fix	EFLAGS, CR0, CR4 and the EXF4 Signal May be Incorrect after Shutdown
AH94	X	X	X	Plan Fix	Performance Monitoring Counter MACRO_INSTS.DECODED May Not Count Some Decoded Instructions
AH95	X	X	X	Plan Fix	The Stack May be Incorrect as a Result of VIP/VIF Check on SYSEXIT and SYSRET
AH96	X	X	X	No Fix	Performance Monitoring Event SIMD_UOP_TYPE_EXEC.MUL is Counted Incorrectly for PMULUDQ Instruction
AH97	X	X	X	No Fix	Storage of PEBS Record Delayed Following Execution of MOV SS or STI

Number	Stepping	Stepping	Stepping	Plans	ERRATA
	B-2	L-2	A-1		
AH98	X	X	X	No Fix	Updating Code Page Directory Attributes without TLB Invalidation May Result in Improper Handling of Code #PF
AH99	X	X		Plan Fix	Performance Monitoring Event CPU_CLK_UNHALTED.REF May Not Count Clock Cycles According to the Processors Operating Frequency
AH100	X	X	X	Plan Fix	Store Ordering May be Incorrect between WC and WP Memory Types
AH101		X	X	Plan Fix	(E)CX May Get Incorrectly Updated When Performing Fast String REP STOS With Large Data Structures
AH102	X	X	X	Plan Fix	Performance Monitoring Event BR_INST_RETIRED May Count CPUID Instructions as Branches
AH103	X	X	X	No Fix	Performance Monitoring Event MISALIGN_MEM_REF May Over Count
AH104	X	X	X	No Fix	A REP STOS/MOVS to a MONITOR/MWAIT Address Range May Prevent Triggering of the Monitoring Hardware
AH105	X	X	X	Plan Fix	False Level One Data Cache Parity Machine-Check Exceptions May be Signaled
AH106	X	X	X	No Fix	A Memory Access May Get a Wrong Memory Type Following a #GP due to WRMSR to an MTRR Mask
AH107	X	X	X	No Fix	PMI While LBR Freeze Enabled May Result in Old/Out-of-date LBR Information
AH108	X	X		Plan Fix	Overlap of an Intel® VT APIC Access Page in a Guest with the DS Save Area May Lead to Unpredictable Behavior
AH109	X	X		No Fix	VTPR Write Access During Event Delivery May Cause an APIC-Access VM Exit
AH110				No Fix	BIST Failure After Reset
AH111	X	X	X	No Fix	Performance Monitoring Event FP_MMX_TRANS_TO_MMX May Not Count Some Transitions
AH112	X	X	X	No Fix	Instruction Fetch May Cause a Livelock During Snoops of the L1 Data Cache
AH113	X	X	X	No Fix	Use of Memory Aliasing with Inconsistent Memory Type may Cause a System Hang or a Machine Check Exception
AH114	X	X	X	No Fix	A WB Store Following a REP STOS/MOVS or FXSAVE May Lead to Memory-Ordering Violations
AH115	X	X	X	No Fix	VM Exit with Exit Reason "TPR Below Threshold" Can Cause the Blocking by MOV/POP SS and Blocking by STI Bits to Be Cleared in the Guest Interruptibility-State Field
AH116	X	X	X	No Fix	Using Memory Type Aliasing with Cacheable and WC Memory Types May Lead to Memory Ordering Violations
AH117	X	X	X	No Fix	RSM Instruction Execution under Certain Conditions May Cause Processor Hang or Unexpected Instruction Execution Results

Number	Stepping	Stepping	Stepping	Plans	ERRATA
	B-2	L-2	A-1		
AH118	X	X	X	No Fix	NMIs may not be blocked by a VM-Entry failure.
AH119	X	X	X	No Fix	Benign Exception after a Double Fault May Not Cause a Triple Fault Shutdown Problem
AH120	X	X	X	No Fix	IA32_MC1_STATUS MSR Bit[60] Does Not Reflect Machine Check Error Reporting Enable Correctly
AH121	X	X	X	No Fix	Corruption of CS Segment Register During RSM While Transitioning From Real Mode to Protected Mode
AH122	X	X	X	No Fix	FP Data Operand Pointer May Be Incorrectly Calculated After an FP Access Which Wraps a 4-Gbyte Boundary in Code That Uses 32-Bit Address Size in 64-bit Mode

Errata for Intel® Core™2 Duo Processors for Platforms Based on Mobile Intel® 965 Express Chipset Family

Number	Stepping	Stepping	Stepping	Plans	ERRATA
	E-1	M-1	G-2		
AH1	X	X	X	No Fix	Writing the Local Vector Table (LVT) When an Interrupt Is Pending May Cause an Unexpected Interrupt
AH2	X	X	X	No Fix	LOCK# Asserted During a Special Cycle Shutdown Transaction May Unexpectedly Deassert
AH3	X	X	X	No Fix	Address Reported by Machine-Check Architecture (MCA) on Single-bit L2 ECC Errors May Be Incorrect
AH4	X	X	X	No Fix	VERW/VERR/LSL/LAR Instructions May Unexpectedly Update the Last Exception Record (LER) MSR
AH5	X	X	X	No Fix	DR3 Address Match on MOVD/MOVQ/MOVNTQ Memory Store Instruction May Incorrectly Increment Performance Monitoring Count for Saturating SIMD Instructions Retired (Event CFH)
AH6				Fixed	SYSRET May Incorrectly Clear RF (Resume Flag) in the RFLAGS Register
AH7	X	X	X	No Fix	General Protection Fault (#GP) for Instructions Greater than 15 Bytes May Be Preempted
AH8	X	X	X	No Fix	Pending x87 FPU Exceptions (#MF) Following STI May Be Serviced Before Higher Priority Interrupts
AH9	X	X	X	No Fix	The Processor May Report a #TS Instead of a #GP Fault
AH10					Removed Erratum
AH11	X	X	X	No Fix	A Write to an APIC Register Sometimes May Appear to Have Not Occurred
AH12	X	X	X	No Fix	Programming the Digital Thermal Sensor (DTS) Threshold May Cause Unexpected Thermal Interrupts
AH13	X	X	X	No Fix	Count Value for Performance-Monitoring Counter PMH_PAGE_WALK May Be Incorrect
AH14	X	X	X	No Fix	LER MSRs May be Incorrectly Updated
AH15	X	X	X	No Fix	Performance Monitoring Events for Retired Instructions (C0H) May Not Be Accurate
AH16	X	X	X	No Fix	Performance Monitoring Event For Number Of Reference Cycles When The Processor Is Not Halted (3CH) Does Not Count According To The Specification
AH17	X	X	X	No Fix	Using 2M/4M Pages When A20M# Is Asserted May Result in Incorrect Address Translations
AH18	X	X	X	No Fix	Writing Shared Unaligned Data that Crosses a Cache Line without Proper Semaphores or Barriers May Expose a Memory Ordering Issue

Number	Stepping	Stepping	Stepping	Plans	ERRATA
	E-1	M-1	G-2		
AH19	X	X	X	No Fix	Code Segment Limit Violation May Occur On 4 Gigabyte Limit Check
AH20				Fixed	FP Inexact-Result Exception Flag May Not Be Set
AH21				Fixed	Global Pages in the Data Translation Look-Aside Buffer (DTLB) May Not Be Flushed by RSM instruction before Restoring the Architectural State from SMRAM
AH22				Fixed	Sequential Code Fetch to Non-canonical Address May have Nondeterministic Results
AH23				Fixed	VMCALL to Activate Dual-monitor Treatment of SMIs and SMM Ignores Reserved Bit settings in VM-exit Control Field
AH24	X	X	X	No Fix	REP MOVS/STOS Executing with Fast Strings Enabled and Crossing Page Boundaries with Inconsistent Memory Types May Use an Incorrect Data Size or Lead to Memory-Ordering Violations.
AH25	X	X	X	No Fix	Some Bus Performance Monitoring Events May Not Count Local Events under Certain Conditions
AH26	X	X	X	No Fix	Premature Execution of a Load Operation Prior to Exception Handler Invocation
AH27	X	X	X	No Fix	General Protection (#GP) Fault May Not Be Signaled on Data Segment Limit Violation above 4-G Limit
AH28	X	X	X	No Fix	EIP May Be Incorrect after Shutdown in IA-32e Mode
AH29	X	X	X	No Fix	#GP Fault is Not Generated on Writing IA32_MISC_ENABLE [34] When Execute Disable is Not supported
AH30					Removed Erratum
AH31				Fixed	Performance Monitoring Events for Retired Loads (CBH) and Instructions Retired (C0H) May Not Be Accurate
AH32	X	X	X	No Fix	Upper 32 bits of 'From' Address Reported through BTMs or BTSs May Be Incorrect
AH33				Fixed	Unsynchronized Cross-Modifying Code Operations Can Cause unexpected Instruction Execution Results
AH34	X	X	X	No Fix	MSRs Actual Frequency Clock Count (IA32_APERF) or Maximum Frequency Clock Count (IA32_MPERF) May Contain Incorrect Data after a Machine Check Exception (MCE)
AH35	X	X	X	No Fix	Incorrect Address Computed for Last Byte of FXSAVE/FXRSTOR Image Leads to Partial Memory Update
AH36	X	X	X	No Fix	Split Locked Stores May Not Trigger the Monitoring Hardware
AH37				Fixed	REP CMPS/SCAS Operations May Terminate Early in 64-bit Mode When RCX >= 0X100000000
AH38				Fixed	FXSAVE/FXRSTOR Instructions which Store to the End of the Segment and Cause a Wrap to a Misaligned Base Address (Alignment <= 0x10h) May Cause FPU Instruction or Operand Pointer Corruption

Number	Stepping	Stepping	Stepping	Plans	ERRATA
	E-1	M-1	G-2		
AH39				Fixed	Cache Data Access Request from One Core Hitting a Modified Line in the L1 Data Cache of the Other Core May Cause Unpredictable System Behavior
AH40				Fixed	PREFETCHh Instruction Execution under Some Conditions May Lead to Processor Livelock
AH41				Fixed	PREFETCHh Instructions May Not Be Executed when Alignment Check (AC) Is Enabled
AH42				Fixed	Upper 32 Bits of the FPU Data (Operand) Pointer in the FXSAVE Memory Image May Be Unexpectedly All 1's after FXSAVE
AH43				Fixed	Concurrent Multi-processor Writes to Non-dirty Page May Result in Unpredictable Behavior
AH44				Fixed	Performance Monitor IDLE_DURING_DIV (18h) Count May Not Be Accurate
AH45	X	X	X	No Fix	Values for LBR/BTS/BTM Will Be Incorrect after an Exit from SMM
AH46	X	X	X	No Fix	Shutdown Condition May Disable Non-Bootstrap Processors
AH47				Fixed	SYSCALL Immediately after Changing EFLAGS.TF May Not Behave According to the New EFLAGS.TF
AH48	X	X	X	No Fix	Code Segment Limit/Canonical Faults on RSM May be Serviced before Higher Priority Interrupts/Exceptions and May Push the Wrong Address Onto the Stack
AH49	X	X	X	No Fix	VM Bit Is Cleared on Second Fault Handled by Task Switch from Virtual-8086 (VM86)
AH50				Fixed	IA32_FMASK Is Reset during an INIT
AH51	X	X	X	No Fix	An Enabled Debug Breakpoint or Single Step Trap May Be Taken after MOV SS/POP SS Instruction if it is Followed by an Instruction That Signals a Floating Point Exception
AH52	X	X	X	No Fix	Last Branch Records (LBR) Updates May Be Incorrect after a Task Switch
AH53	X	X	X	No Fix	IO_SMI Indication in SMRAM State Save Area May Be Set Incorrectly
AH54	X	X	X	No Fix	INIT Does Not Clear Global Entries in the TLB
AH55				Fixed	Using Memory Type Aliasing with Memory Types WB/WT May Lead to Unpredictable Behavior
AH56				Fixed	Update of Read/Write (R/W) or User/Supervisor (U/S) or Present (P) Bits without TLB Shootdown May Cause Unexpected Processor Behavior
AH57				Fixed	BTS Message May Be Lost When the STPCLK# Signal Is Active
AH58	X	X	X	No Fix	MOV To/From Debug Registers Causes Debug Exception
AH59	X	X	X	No Fix	EFLAGS Discrepancy on a Page Fault after a Multiprocessor TLB Shootdown

Number	Stepping	Stepping	Stepping	Plans	ERRATA
	E-1	M-1	G-2		
AH60	X	X	X	No Fix	LBR, BTS, BTM May Report a Wrong Address when an Exception/Interrupt Occurs in 64-bit Mode
AH61	X	X	X	No Fix	A Thermal Interrupt Is Not Generated when the Current Temperature Is Invalid
AH62	X	X	X	No Fix	CMPSB, LODSB, or SCASB in 64-bit Mode with Count Greater or Equal to 2^{48} May Terminate Early
AH63					Removed Erratum
AH64	X	X	X	No Fix	Returning to Real Mode from SMM with EFLAGS.VM Set May Result in Unpredictable System Behavior
AH65				Fixed	VMLAUNCH/VMRESUME May Not Fail When VMCS Is Programmed to Cause VM Exit to Return to a Different Mode
AH66	X	X	X	No Fix	IRET under Certain Conditions May Cause an Unexpected Alignment Check Exception
AH67	X	X	X	No Fix	Performance Monitoring Event FP_ASSIST May Not Be Accurate
AH68				Fixed	CPL-Qualified BTS May Report Incorrect Branch-From Instruction Address
AH69				Fixed	PEBS Does Not Always Differentiate Between CPL-Qualified Events
AH70	X	X	X	No Fix	PMI May Be Delayed to Next PEBS Event
AH71				Fixed	PEBS Buffer Overflow Status Will Not Be Indicated Unless IA32_DEBUGCTL[12] Is Set
AH72	X	X	X	No Fix	The BS Flag in DR6 May Be Set for Non-Single-Step #DB Exception
AH73	X	X	X	No Fix	An Asynchronous MCE during a Far Transfer May Corrupt ESP
AH74				Fixed	In Single-Stepping on Branches Mode, the BS Bit in the Pending-Debug-Exceptions Field of the Guest State Area will be Incorrectly Set by VM Exit on a MOV to CR8 Instruction
AH75	X	X	X	No Fix	B0-B3 Bits in DR6 May Not Be Properly Cleared after Code Breakpoint
AH76	X	X	X	No Fix	BTM/BTS Branch-From Instruction Address May Be Incorrect for Software Interrupts
AH77				Fixed	REP Store Instructions in a Specific Situation May Cause the Processor to Hang
AH78	X	X	X	No Fix	Performance Monitor SSE Retired Instructions May Return Incorrect Values
AH79	X	X	X	No Fix	Performance Monitoring Events for L1 and L2 Miss May Not Be Accurate
AH80	X	X	X	No Fix	Store to WT Memory Data May Be Seen in Wrong Order by Two Subsequent Loads
AH81	X	X	X	No Fix	A MOV Instruction from CR8 Register with 16 Bit Operand Size Will Leave Bits 63:16 of the Destination Register Unmodified

Number	Stepping	Stepping	Stepping	Plans	ERRATA
	E-1	M-1	G-2		
AH82				Fixed	Debug Register May Contain Incorrect Information on a MOVSS or POPSS Instruction followed by SYSRET
AH83	X	X	X	No Fix	Single Step Interrupts with Floating Point Exception Pending May Be Mishandled
AH84	X	X	X	No Fix	Non-Temporal Data Store May Be Observed in Wrong Program Order
AH85	X	X	X	No Fix	Fault on ENTER Instruction May Result in Unexpected Values on Stack Frame
AH86	X	X		Fixed	CPUID Reports Architectural Performance Monitoring Version 2 is Supported, When Only Version 1 Capabilities are Available
AH87	X	X	X	No Fix	Unaligned Accesses to Paging Structures May Cause the Processor to Hang
AH88	X	X	X	No Fix	Microcode Updates Performed During VMX Non-root Operation Could Result in Unexpected Behavior
AH89	X	X	X	No Fix	INVLPG Operation for Large (2M/4M) Pages May be Incomplete under Certain Conditions
AH90	X	X	X	No Fix	Page Access Bit May be Set Prior to Signaling a Code Segment Limit Fault
AH91	X	X		Fixed	Update of Attribute Bits on Page Directories without Immediate TLB Shootdown May Cause Unexpected Processor Behavior
AH92				Fixed	Invalid Instructions May Lead to Unexpected Behavior
AH93	X	X	X	No Fix	EFLAGS, CR0, CR4 and the EXF4 Signal May be Incorrect after Shutdown
AH94				Fixed	Performance Monitoring Counter MACRO_INSTS.DECODED May Not Count Some Decoded Instructions
AH95	X	X		Fixed	The Stack May be Incorrect as a Result of VIP/VIF Check on SYSEXIT and SYSRET
AH96	X	X	X	No Fix	Performance Monitoring Event SIMD_UOP_TYPE_EXEC.MUL is Counted Incorrectly for PMULUDQ Instruction
AH97	X	X	X	No Fix	Storage of PEBS Record Delayed Following Execution of MOV SS or STI
AH98	X	X	X	No Fix	Updating Code Page Directory Attributes without TLB Invalidation May Result in Improper Handling of Code #PF
AH99	X	X		Fixed	Performance Monitoring Event CPU_CLK_UNHALTED.REF May Not Count Clock Cycles According to the Processors Operating Frequency
AH100	X	X	X	Plan Fix	Store Ordering May be Incorrect between WC and WP Memory Types
AH101				Fixed	(E)CX May Get Incorrectly Updated When Performing Fast String REP STOS With Large Data Structures

Number	Stepping	Stepping	Stepping	Plans	ERRATA
	E-1	M-1	G-2		
AH102				Fixed	Performance Monitoring Event BR_INST_RETIRED May Count CPUID Instructions as Branches
AH103	X	X	X	No Fix	Performance Monitoring Event MISALIGN_MEM_REF May Over Count
AH104	X	X	X	No Fix	A REP STOS/MOVS to a MONITOR/MWAIT Address Range May Prevent Triggering of the Monitoring Hardware
AH105				Fixed	False Level One Data Cache Parity Machine-Check Exceptions May be Signaled
AH106	X	X	X	No Fix	A Memory Access May Get a Wrong Memory Type Following a #GP due to WRMSR to an MTRR Mask
AH107	X	X	X	No Fix	PMI While LBR Freeze Enabled May Result in Old/Out-of-date LBR Information
AH108	X	X		Fixed	Overlap of an Intel® VT APIC Access Page in a Guest with the DS Save Area May Lead to Unpredictable Behavior
AH109	X	X	X	No Fix	VTPR Write Access During Event Delivery May Cause an APIC-Access VM Exit
AH110	X	X	X	No Fix	BIST Failure After Reset
AH111	X	X	X	No Fix	Performance Monitoring Event FP_MMX_TRANS_TO_MMX May Not Count Some Transitions
AH112	X	X	X	No Fix	Instruction Fetch May Cause a Livelock during Snoops of the L1 Data Cache
AH113	X	X	X	No Fix	Use of Memory Aliasing with Inconsistent Memory Type may Cause a System Hang or a Machine Check Exception
AH114	X	X	X	No Fix	A WB Store Following a REP STOS/MOVS or FXSAVE May Lead to Memory-Ordering Violations
AH115	X	X	X	No Fix	VM Exit with Exit Reason “TPR Below Threshold” Can Cause the Blocking by MOV/POP SS and Blocking by STI Bits to Be Cleared in the Guest Interruptibility-State Field
AH116	X	X	X	No Fix	Using Memory Type Aliasing with Cacheable and WC Memory Types May Lead to Memory Ordering Violations
AH117	X	X	X	No Fix	RSM Instruction Execution under Certain Conditions May Cause Processor Hang or Unexpected Instruction Execution Results
AH118	X	X	X	No Fix	NMIs may not be blocked by a VM-Entry failure.
AH119	X	X	X	No Fix	Benign Exception after a Double Fault May Not Cause a Triple Fault Shutdown Problem
AH120	X	X	X	No Fix	IA32_MC1_STATUS MSR Bit[60] Does Not Reflect Machine Check Error Reporting Enable Correctly
AH121	X	X	X	No Fix	Corruption of CS Segment Register During RSM While Transitioning From Real Mode to Protected Mode
AH122	X	X	X	No Fix	FP Data Operand Pointer May Be Incorrectly Calculated After an FP Access Which Wraps a 4-Gbyte Boundary in Code That Uses 32-Bit Address Size in 64-bit Mode

Number	Stepping E-1	Stepping M-1	Stepping G-2	Plans	ERRATA
Errata Affecting Only Intel® Core™2 Duo Mobile Processor on Mobile Intel® 965 Express Chipset Family					
AH1P	X	X		Plan Fix	VM Exit due to Virtual APIC-Access May Clear RF
AH2P	X	X		Fixed	VMCALL Failure Due to Corrupt MSEG Location May Cause VM Exit to Load the Machine State Incorrectly
AH3P	X	X		Fixed	Fixed Function Performance Counters MSR_PERF_FIXED_CTR1 (30AH) and MSR_PERF_FIXED (30BH) are Not Cleared When the Processor is Reset
AH4P	X	X	X	No Fix	Multi-Core Processors Configured for Single Core Operation May Not be Able to Enter Intel® Enhanced Deeper Sleep
AH5P	X	X		Fixed	VTPR Access May Lead to a System Hang
AH6P	X		X	No Fix	Activation of Intel® Adaptive Thermal Monitor While Intel® Dynamic Front Side Bus Frequency Switching is Active May lead to an Incorrect Operating Point Frequency

NOTES:

1. Errata with a "P" designation are specific to the Intel Core 2 Duo Mobile Processor for the Mobile Intel 965 Express Chipset Family.

Errata for Intel® Celeron® Processor 500 Series for Platforms Based on Mobile Intel® 965 Express Chipset Family

Number	Stepping	Stepping	Stepping	Plans	Errata
	A-1	E-1	M-1		
AH1	X	X	X	No Fix	Writing the Local Vector Table (LVT) When an Interrupt Is Pending May Cause an Unexpected Interrupt
AH2	X	X	X	No Fix	LOCK# Asserted During a Special Cycle Shutdown Transaction May Unexpectedly Deassert
AR3				No Fix	Erratum Removed
AH4	X	X	X	No Fix	VERW/VERR/LSL/LAR Instructions May Unexpectedly Update the Last Exception Record (LER) MSR
AH5	X	X	X	No Fix	DR3 Address Match on MOVD/MOVQ/MOVNTQ Memory Store Instruction May Incorrectly Increment Performance Monitoring Count for Saturating SIMD Instructions Retired (Event CFH)
AH6	X			Fixed	SYSRET May Incorrectly Clear RF (Resume Flag) in the RFLAGS Register
AH7	X	X	X	No Fix	General Protection Fault (#GP) for Instructions Greater than 15 Bytes May Be Preempted
AH8	X	X	X	No Fix	Pending x87 FPU Exceptions (#MF) Following STI May Be Serviced Before Higher Priority Interrupts
AH9	X	X	X	No Fix	The Processor May Report a #TS Instead of a #GP Fault
AH11	X	X	X	No Fix	A Write to an APIC Register Sometimes May Appear to Have Not Occurred
AH12	X	X	X	No Fix	Programming the Digital Thermal Sensor (DTS) Threshold May Cause Unexpected Thermal Interrupts
AH13	X	X	X	No Fix	Count Value for Performance-Monitoring Counter PMH_PAGE_WALK May Be Incorrect
AH14	X	X	X	No Fix	LER MSRs May Be Incorrectly Updated
AH15	X	X	X	No Fix	Performance Monitoring Events for Retired Instructions (C0H) May Not Be Accurate
AH16	X	X	X	No Fix	Performance Monitoring Event For Number Of Reference Cycles When The Processor Is Not Halted (3CH) Does Not Count According To The Specification
AR16		X		Fixed	Using 2M/4M Pages When A20M# Is Asserted May Result in Incorrect Address Translations
AH18	X	X	X	No Fix	Writing Shared Unaligned Data that Crosses a Cache Line without Proper Semaphores or Barriers May Expose a Memory Ordering Issue
AH19	X	X	X	No Fix	Code Segment Limit Violation May Occur On 4 Gigabyte Limit Check
AH20	x			Fixed	FP Inexact-Result Exception Flag May Not Be Set

Number	Stepping	Stepping	Stepping	Plans	Errata
	A-1	E-1	M-1		
AH21	X			Fixed	Global Pages in the Data Translation Look-Aside Buffer (DTLB) May Not Be Flushed by RSM instruction before Restoring the Architectural State from SMRAM
AH22	X			Fixed	Sequential Code Fetch to Non-canonical Address May have Nondeterministic Results
AH24	X	X	X	No Fix	REP MOVS/STOS Executing with Fast Strings Enabled and Crossing Page Boundaries with Inconsistent Memory Types May Use an Incorrect Data Size or Lead to Memory-Ordering Violations.
AH25	X	X	X	No Fix	Some Bus Performance Monitoring Events May Not Count Local Events under Certain Conditions
AH26	X	X	X	No Fix	Premature Execution of a Load Operation Prior to Exception Handler Invocation
AH27	X	X	X	No Fix	General Protection (#GP) Fault May Not Be Signaled on Data Segment Limit Violation above 4-G Limit
AH28	X	X	X	No Fix	EIP May Be Incorrect after Shutdown in IA-32e Mode
AH29	X	X	X	No Fix	#GP Fault Is Not Generated on Writing IA32_MISC_ENABLE [34] When Execute Disable Is Not supported
AH30	X			Fixed	(E)CX May Get Incorrectly Updated Fast String REP MOVS or Fast String REP STOS with Large Data Structures
AH31	X			Fixed	Performance Monitoring Events for Retired Loads (CBH) and Instructions Retired (C0H) May Not Be Accurate
AH32	X	X	X	No Fix	Upper 32 bits of 'From' Address Reported through BTMs or BTSs May Be Incorrect
AH33	X			Fixed	Unsynchronized Cross-Modifying Code Operations Can Cause unexpected Instruction Execution Results
AH34	X	X	X	No Fix	MSRs Actual Frequency Clock Count (IA32_APERF) or Maximum Frequency Clock Count (IA32_MPERF) May Contain Incorrect Data after a Machine Check Exception (MCE)
AH35	X	X	X	No Fix	Incorrect Address Computed for Last Byte of FXSAVE/FXRSTOR Image Leads to Partial Memory Update
AH36	X	X	X	No Fix	Split Locked Stores May Not Trigger the Monitoring Hardware
AH37	X			Fixed	REP CMPS/SCAS Operations May Terminate Early in 64-bit Mode When RCX >= 0X100000000
AH38	X			Fixed	FXSAVE/FXRSTOR Instructions which Store to the End of the Segment and Cause a Wrap to a Misaligned Base Address (Alignment <=0x10h) May Cause FPU Instruction or Operand Pointer Corruption
AH39	X			Fixed	Cache Data Access Request from One Core Hitting a Modified Line in the L1 Data Cache of the Other Core May Cause Unpredictable System Behavior
AH40	X			Fixed	PREFETCHh Instruction Execution under Some Conditions May Lead to Processor Livelock

Number	Stepping	Stepping	Stepping	Plans	Errata
	A-1	E-1	M-1		
AH41	X			Fixed	PREFETCHh Instructions May Not Be Executed when Alignment Check (AC) Is Enabled
AH42	X			Fixed	Upper 32 Bits of the FPU Data (Operand) Pointer in the FXSAVE Memory Image May Be Unexpectedly All 1's after FXSAVE
AH43				Fixed	Concurrent Multi-processor Writes to Non-dirty Page May Result in Unpredictable Behavior
AH44	X			Fixed	Performance Monitor IDLE_DURING_DIV (18h) Count May Not Be Accurate
AH45	X	X	X	No Fix	Values for LBR/BTS/BTM Will Be Incorrect after an Exit from SMM
AH47	X			Fixed	SYSCALL Immediately after Changing EFLAGS.TF May Not Behave According to the New EFLAGS.TF
AH49	X	X	X	No Fix	VM Bit Is Cleared on Second Fault Handled by Task Switch from Virtual-8086 (VM86)
AH50	X			Fixed	IA32_FMASK Is Reset during an INIT
AH51	X	X	X	No Fix	An Enabled Debug Breakpoint or Single Step Trap May Be Taken after MOV SS/POP SS Instruction if it is Followed by an Instruction That Signals a Floating Point Exception
AH52	X	X	X	No Fix	Last Branch Records (LBR) Updates May Be Incorrect after a Task Switch
AH53	X	X	X	No Fix	IO_SMI Indication in SMRAM State Save Area May Be Set Incorrectly
AH54	X	X	X	No Fix	INIT Does Not Clear Global Entries in the TLB
AH55	X			Fixed	Using Memory Type Aliasing with Memory Types WB/WT May Lead to Unpredictable Behavior
AH56	X			Fixed	Update of Read/Write (R/W) or User/Supervisor (U/S) or Present (P) Bits without TLB Shootdown May Cause Unexpected Processor Behavior
AH57	X			Fixed	BTS Message May Be Lost When the STPCLK# Signal Is Active
AH58	X	X	X	No Fix	MOV To/From Debug Registers Causes Debug Exception
AH59	X	X	X	No Fix	EFLAGS Discrepancy on a Page Fault after a Multiprocessor TLB Shootdown
AH60	X	X	X	No Fix	LBR, BTS, BTM May Report a Wrong Address when an Exception/Interrupt Occurs in 64-bit Mode
AH61	X	X	X	No Fix	A Thermal Interrupt Is Not Generated when the Current Temperature Is Invalid
AH62	X	X	X	No Fix	CMPSB, LODSB, or SCASB in 64-bit Mode with Count Greater or Equal to 2^{48} May Terminate Early
AH64	X	X	X	No Fix	Returning to Real Mode from SMM with EFLAGS.VM Set May Result in Unpredictable System Behavior
AH66	X	X	X	No Fix	IRET under Certain Conditions May Cause an Unexpected Alignment Check Exception

Number	Stepping	Stepping	Stepping	Plans	Errata
	A-1	E-1	M-1		
AH67	X	X	X	No Fix	Performance Monitoring Event FP_ASSIST May Not Be Accurate
AH68	X			Fixed	CPL-Qualified BTS May Report Incorrect Branch-From Instruction Address
AH69	X			Fixed	PEBS Does Not Always Differentiate Between CPL-Qualified Events
AH70	X	X	X	No Fix	PMI May Be Delayed to Next PEBS Event
AH71	X			Fixed	PEBS Buffer Overflow Status Will Not Be Indicated Unless IA32_DEBUGCTL[12] Is Set
AH72	X	X	X	No Fix	The BS Flag in DR6 May Be Set for Non-Single-Step #DB Exception
AH73	X	X	X	No Fix	An Asynchronous MCE during a Far Transfer May Corrupt ESP
AH75	X	X	X	No Fix	B0-B3 Bits in DR6 May Not Be Properly Cleared after Code Breakpoint
AH76	X	X	X	No Fix	BTM/BTS Branch-From Instruction Address May Be Incorrect for Software Interrupts
AH77	X			Fixed	REP Store Instructions in a Specific Situation May Cause the Processor to Hang
AH78	X	X	X	No Fix	Performance Monitor SSE Retired Instructions May Return Incorrect Values
AH79	X	X	X	No Fix	Performance Monitoring Events for L1 and L2 Miss May Not Be Accurate
AH80				No Fix	Erratum Removed
AH81	X	X	X	No Fix	A MOV Instruction from CR8 Register with 16 Bit Operand Size Will Leave Bits 63:16 of the Destination Register Unmodified
AH82	X			Fixed	Debug Register May Contain Incorrect Information on a MOVSS or POPSS Instruction followed by SYSRET
AH83	X	X	X	No Fix	Single Step Interrupts with Floating Point Exception Pending May Be Mishandled
AH85	X	X	X	No Fix	Fault on ENTER Instruction May Result in Unexpected Values on Stack Frame
AH87	X	X	X	No Fix	Unaligned Accesses to Paging Structures May Cause the Processor to Hang
AH89	X	X	X	No Fix	INVLPG Operation for Large (2M/4M) Pages May Be Incomplete under Certain Conditions
AH90	X	X	X	No Fix	Page Access Bit May Be Set Prior to Signaling a Code Segment Limit Fault
AH91	X	X	X	Plan Fix	Update of Attribute Bits on Page Directories without Immediate TLB Shootdown May Cause Unexpected Processor Behavior
AH92	X			Fixed	Invalid Instructions May Lead to Unexpected Behavior
AH93	X	X	X	No Fix	EFLAGS, CR0, CR4 and the EXF4 Signal May Be Incorrect after Shutdown

Number	Stepping	Stepping	Stepping	Plans	Errata
	A-1	E-1	M-1		
AH94	X			Fixed	Performance Monitoring Counter MACRO_INSTS.DECODED May Not Count Some Decoded Instructions
AH95	X	X	X	Plan Fix	The Stack May Be Incorrect as a Result of VIP/VIF Check on SYSEXIT and SYSRET
AH96	X	X	X	No Fix	Performance Monitoring Event SIMD_UOP_TYPE_EXEC.MUL is Counted Incorrectly for PMULUDQ Instruction
AH97	X	X	X	No Fix	Storage of PEBS Record Delayed Following Execution of MOV SS or STI
AH98	X	X	X	No Fix	Updating Code Page Directory Attributes without TLB Invalidation May Result in Improper Handling of Code #PF
AH99	X	X	X	Plan Fix	Performance Monitoring Event CPU_CLK_UNHALTED.REF May Not Count Clock Cycles According to the Processors Operating Frequency
AH100	X	X	X	Plan Fix	Store Ordering May Be Incorrect between WC and WP Memory Types
AH102	X			Fixed	Performance Monitoring Event BR_INST_RETIRED May Count CPUID Instructions as Branches
AH103	X	X	X	No Fix	Performance Monitoring Event MISALIGN_MEM_REF May Over Count
AH104	X	X	X	No Fix	A REP STOS/MOVS to a MONITOR/MWAIT Address Range May Prevent Triggering of the Monitoring Hardware
AH105	X			Fixed	False Level One Data Cache Parity Machine-Check Exceptions May Be Signaled
AH106	X	X	X	No Fix	A Memory Access May Get a Wrong Memory Type Following a #GP due to WRMSR to an MTRR Mask
AH107	X	X	X	No Fix	PMI While LBR Freeze Enabled May Result in Old/Out-of-date LBR Information
AH110		X	X	No Fix	BIST Failure after Reset
AH111	X	X	X	No Fix	Performance Monitoring Event FP_MMX_TRANS_TO_MMX May Not Count Some Transitions
AH112	X	X	X	No Fix	Instruction Fetch May Cause a Livelock During Snoops of the L1 Data Cache
AH113	X	X	X	No Fix	Use of Memory Aliasing with Inconsistent Memory Type may Cause a System Hang or a Machine Check Exception
AH114	X	X	X	No Fix	A WB Store Following a REP STOS/MOVS or FXSAVE May Lead to Memory-Ordering Violations
AH116	X	X	X	No Fix	Using Memory Type Aliasing with Cacheable and WC Memory Types May Lead to Memory Ordering Violations
AH117	X	X	X	No Fix	RSM Instruction Execution under Certain Conditions May Cause Processor Hang or Unexpected Instruction Execution Results
AH118	X	X	X	No Fix	NMIs May Not Be Blocked by a VM-Entry Failure

Number	Stepping A-1	Stepping E-1	Stepping M-1	Plans	Errata
AH119	X	X	X	No Fix	Benign Exception after a Double Fault May Not Cause a Triple Fault Shutdown Problem
AH120	X	X	X	No Fix	IA32_MC1_STATUS MSR Bit[60] Does Not Reflect Machine Check Error Reporting Enable Correctly
AH48	X	X	X	No Fix	Code Segment Limit/Canonical Faults on RSM May be Serviced before Higher Priority Interrupts/Exceptions and May Push the Wrong Address Onto the Stack
AH122	X	X	X	No Fix	FP Data Operand Pointer May Be Incorrectly Calculated After an FP Access Which Wraps a 4-Gbyte Boundary in Code That Uses 32-Bit Address Size in 64-bit Mode

Number	SPECIFICATION CHANGES
	There are no Specification Changes in this Specification Update revision

Number	SPECIFICATION CLARIFICATIONS
AH3	Clarification of TRANSLATION LOOKASIDE BUFFERS (TLBS) Invalidation

Number	DOCUMENTATION CHANGES
	There are no Documentation Changes in this Specification Update revision.

Errata

AH1. Writing the Local Vector Table (LVT) When an Interrupt Is Pending May Cause an Unexpected Interrupt

Problem: If a local interrupt is pending when the LVT entry is written, an interrupt may be taken on the new interrupt vector even if the mask bit is set.

Implication: An interrupt may immediately be generated with the new vector when a LVT entry is written, even if the new LVT entry has the mask bit set. If there is no Interrupt Service Routine (ISR) set up for that vector the system will GP fault. If the ISR does not do an End of Interrupt (EOI) the bit for the vector will be left set in the in-service register and mask all interrupts at the same or lower priority.

Workaround: Any vector programmed into an LVT entry must have an ISR associated with it, even if that vector was programmed as masked. This ISR routine must do an EOI to clear any unexpected interrupts that may occur. The ISR associated with the spurious vector does not generate an EOI, therefore the spurious vector should not be used when writing the LVT.

Status: For the steppings affected, see the Summary Tables of Changes.

AH2. LOCK# Asserted During a Special Cycle Shutdown Transaction May Unexpectedly Deassert

Problem: During a processor shutdown transaction, when LOCK# is asserted and if a DEFER# is received during a snoop phase and the Locked transaction is pipelined on the front side bus (FSB), LOCK# may unexpectedly deassert.

Implication: When this erratum occurs, the system may hang during shutdown. Intel has not observed this erratum with any commercially available systems or software.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH3 Address Reported by Machine-Check Architecture (MCA) on Single-bit L2 ECC Errors May Be Incorrect

Problem: When correctable Single-bit ECC errors occur in the L2 cache, the address is logged in the MCA address register (MCi_ADDR). Under some scenarios, the address reported may be incorrect.

Implication: Software should not rely on the value reported in MCi_ADDR, for Single-bit L2 ECC errors.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH4. Exception Record (LER) MSRVERW/VERR/LSL/LAR Instructions May Unexpectedly Update the Last Exception Record (LER) MSR

Problem: The LER MSR may be unexpectedly updated, if the resultant value of the Zero Flag (ZF) is zero after executing the following instructions.

1. VERR (ZF=0 indicates unsuccessful segment read verification)
2. VERW (ZF=0 indicates unsuccessful segment write verification)
3. LAR (ZF=0 indicates unsuccessful access rights load)
4. LSL (ZF=0 indicates unsuccessful segment limit load)

Implication: The value of the LER MSR may be inaccurate if VERW/VERR/LSL/LAR instructions are executed after the occurrence of an exception.

Workaround: Software exception handlers that rely on the LER MSR value should read the LER MSR before executing VERW/VERR/LSL/LAR instructions.

Status: For the steppings affected, see the Summary Tables of Changes.

AH5. DR3 Address Match on MOVD/MOVQ/MOVNTQ Memory Store Instruction May Incorrectly Increment Performance Monitoring Count for Saturating SIMD Instructions Retired (Event CFH)

Problem: Performance monitoring for Event CFH normally increments on saturating SIMD instruction retired. Regardless of DR7 programming, if the linear address of a retiring memory store MOVD/MOVQ/MOVNTQ instruction executed matches the address in DR3, the CFH counter may be incorrectly incremented.

Implication: The value observed for performance monitoring count for saturating SIMD instructions retired may be too high. The size of the error is dependent on the number of occurrences of the conditions described above, while the counter is active.

None Identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH6. SYSRET May Incorrectly Clear RF (Resume Flag) in the RFLAGS Register

Problem: In normal operation, SYSRET will restore the value of RFLAGS from R11 (the value previously saved upon execution of the SYSCALL instruction). Due to this erratum, the RFLAGS.RF bit will be unconditionally cleared after execution of the SYSRET instruction.

Implication: The SYSRET instruction can not be used if the RF flag needs to be set after returning from a system call. Intel has not observed this erratum with any commercially available software.

Workaround: Use the IRET instruction to return from a system call, if RF flag has to be set after the return.

Status: For the steppings affected, see the Summary Tables of Changes

AH7. General Protection Fault (#GP) for Instructions Greater Than 15 Bytes May Be Preempted

Problem: When the processor encounters an instruction that is greater than 15 bytes in length, a #GP is signaled when the instruction is decoded. Under some circumstances, the #GP fault may be preempted by another lower priority fault (for example, Page Fault (#PF)). However, if the preempting lower priority faults are resolved by the operating system and the instruction retried, a #GP fault will occur.

Implication: Software may observe a lower-priority fault occurring before or in lieu of a #GP fault. Instructions of greater than 15 bytes in length can only occur if redundant prefixes are placed before the instruction.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH8. Pending x87 FPU exceptions (#MF) following STI may be serviced before higher priority interrupts.

Problem: Interrupts that are pending prior to the execution of the STI (Set Interrupt Flag) instruction are serviced immediately after the STI instruction is executed. Because of this erratum, if following STI, an instruction that triggers a #MF is executed while STPCLK#, Enhanced Intel SpeedStep® Technology transitions or Thermal Monitor 1 events occur, the pending #MF may be serviced before higher priority interrupts.

Implication: Software may observe #MF being serviced before higher priority interrupts.

Workaround: None Identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH9. The Processor May Report a #TS Instead of a #GP Fault

Problem: A jump to a busy TSS (Task-State Segment) may cause a #TS (invalid TSS exception) instead of a #GP fault (general protection exception).

Implication: Operation systems that access a busy TSS may get invalid TSS fault instead of a #GP fault. Intel has not observed this erratum with any commercially available software.

Workaround: None Identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH10. Removed Erratum

AH11. A Write to an APIC Register Sometimes May Appear to Have Not Occurred

Problem: With respect to the retirement of instructions, stores to the uncacheable memory-based APIC register space are handled in a non-synchronized way. For example if an instruction that masks the interrupt flag, for example. CLI, is executed soon after an uncacheable write to the Task Priority Register (TPR) that lowers the APIC priority, the interrupt masking operation may take effect before the actual priority has been lowered. This may cause interrupts whose priority is lower than the initial TPR, but higher than the final TPR, to not be serviced until the interrupt enabled flag is finally set, that is. by STI instruction. Interrupts will remain pending and are not lost.

Implication: In this example the processor may allow interrupts to be accepted but may delay their service.

Workaround: This non-synchronization can be avoided by issuing an APIC register read after the APIC register write. This will force the store to the APIC register before any subsequent instructions are executed. No commercial operating system is known to be impacted by this erratum.

Status: For the steppings affected, see the Summary Tables of Changes.

AH12. Programming the Digital Thermal Sensor (DTS) Threshold May Cause Unexpected Thermal Interrupts

Problem: Software can enable DTS thermal interrupts by programming the thermal threshold and setting the respective thermal interrupt enable bit. When programming DTS value, the previous DTS threshold may be crossed. This generates an unexpected thermal interrupt.

Implication: Software may observe an unexpected thermal interrupt occur after reprogramming the thermal threshold.

Workaround: In the ACPI/OS implement a workaround by temporarily disabling the DTS threshold interrupt before updating the DTS threshold value.

Status: For the steppings affected, see the Summary Tables of Changes.

AH13. Count Value for Performance-Monitoring Counter PMH_PAGE_WALK May Be Incorrect

Problem: Performance-Monitoring Counter PMH_PAGE_WALK is used to count the number of page walks resulting from Data Translation Look-Aside Buffer (DTLB) and Instruction Translation Look-Aside (ITLB) misses. Under certain conditions, this counter may be incorrect.

Implication: There may be small errors in the accuracy of the counter.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH14. LER MSRs May Be Incorrectly Updated

Problem: The LER (Last Exception Record) MSRs, MSR_LER_FROM_LIP (1DDH) and MSR_LER_TO_LIP (1DEH) may contain incorrect values after any of the following:

- Either STPCLK#, NMI (Non-Maskable Interrupt), or external interrupts
- CMP or TEST instructions with an uncacheable memory operand followed by a conditional jump.
- STI/POP SS/MOV SS instructions followed by CMP or TEST instructions and then by a conditional jump.

Implication: When the conditions for this erratum occur, the value of the LER MSRs may be incorrectly updated.

Workaround: None Identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH15. Performance Monitoring Events for Retired Instructions (C0H) May Not Be Accurate

Problem: The INST_RETIRED performance monitor may miscount retired instructions as follows:

- Repeat string and repeat I/O operations are not counted when a hardware interrupt is received during or after the last iteration of the repeat flow.
- VMLAUNCH and VMRESUME instructions are not counted.
- HLT and MWAIT instructions are not counted.

The following instructions, if executed during HLT or MWAIT events, are also not counted:

5. RSM from a C-state SMI during an MWAIT instruction.
6. RSM from an SMI during a HLT instruction.

Implication: There may be a smaller than expected value in the INST_RETIRED performance monitoring counter. The extent to which this value is smaller than expected is determined by the frequency of the above cases.

Workaround: None Identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH16. Performance Monitoring Event for Number of Reference Cycles When the Processor Is Not Halted (3CH) Does Not Count According to the Specification

Problem: The CPU_CLK_UNHALTED performance monitor with mask 1 counts bus clock cycles instead of counting the core clock cycles at the maximum possible ratio. The maximum possible ratio is computed by dividing the maximum possible core frequency by the bus frequency.

Implication: The CPU_CLK_UNHALTED performance monitor with mask 1 counts a value lower than expected. The value is lower by exactly one multiple of the maximum possible ratio.

Workaround: Multiply the performance monitor value by the maximum possible ratio.

Status: For the steppings affected, see the Summary Tables of Changes.

AH17. Using 2M/4M Pages When A20M# Is Asserted May Result in Incorrect Address Translations

Problem: An external A20M# pin if enabled forces address bit 20 to be masked (forced to zero) to emulates real-address mode address wraparound at 1 megabyte. However, if all of the following conditions are met, address bit 20 may not be masked:

- Paging is enabled
- A linear address has bit 20 set
- The address references a large page
- A20M# is enabled

Implication: When A20M# is enabled and an address references a large page the resulting translated physical address may be incorrect. This erratum has not been observed with any commercially available operating system.

Workaround: Operating systems should not allow A20M# to be enabled if the masking of address bit 20 could be applied to an address that references a large page. A20M# is normally only used with the first megabyte of memory.

Status: For the steppings affected, see the Summary Tables of Changes.

AH18. Writing Shared Unaligned Data that Crosses a Cache Line without Proper Semaphores or Barriers May Expose a Memory Ordering Issue

Problem: Software which is written so that multiple agents can modify the same shared unaligned memory location at the same time may experience a memory ordering issue if multiple loads access this shared data shortly thereafter. Exposure to this problem requires the use of a data write which spans a cache line boundary.

Implication: This erratum may cause loads to be observed out of order. Intel has not observed this erratum with any commercially available software or system.

Workaround: Software should ensure at least one of the following is true when modifying shared data by multiple agents:

- The shared data is aligned.
- Proper semaphores or barriers are used in order to prevent concurrent data accesses.

Status: For the steppings affected, see the Summary Tables of Changes.

AH19. Code Segment Limit Violation May Occur On 4-Gigabyte Limit Check

Problem: Code Segment limit violation may occur on 4-Gigabyte limit check when the code stream wraps around in a way that one instruction ends at the last byte of the segment and the next instruction begins at 0x0.

Implication: This is a rare condition that may result in a system hang. Intel has not observed this erratum with any commercially available software, or system.

Workaround: Avoid code that wraps around segment limit.

Status: For the steppings affected, see the Summary Tables of Changes.

AH20. FP Inexact-Result Exception Flag May Not Be Set

Problem: When the result of a floating-point operation is not exactly representable in the destination format (1/3 in binary form, for example), an inexact-result (precision) exception occurs. When this occurs, the PE bit (bit 5 of the FPU status word) is normally set by the processor. Under certain rare conditions, this bit may not be set when this rounding occurs. However, other actions taken by the processor (invoking the software exception handler if the exception is unmasked) are not affected. This erratum can only occur if one of the following FST instructions is one or two instructions after the floating-point operation which causes the precision exception:

- FST m32real
- FST m64real
- FSTP m32real
- FSTP m64real
- FSTP m80real
- FIST m16int
- FIST m32int
- FISTP m16int
- FISTP m32int
- FISTP m64int
- FISTTP m16int
- FISTTP m32int
- FISTTP m64int

Note: Even if this combination of instructions is encountered, there is also a dependency on the internal pipelining and execution state of both instructions in the processor.

Implication: Inexact-result exceptions are commonly masked or ignored by applications, as it happens frequently, and produces a rounded result acceptable to most applications. The PE bit of the FPU status word may not always be set upon receiving an inexact-result exception. Thus, if these exceptions are unmasked, a floating-point error exception handler may not recognize that a precision exception occurred. Note that this is a "sticky" bit, i.e., once set by an inexact-result condition, it remains set until cleared by software.

Workaround: This condition can be avoided by inserting either three NOPs or three non-floating-point non-Jcc instructions between the two floating-point instructions.

Status: For the steppings affected, see the Summary Tables of Changes.

AH21. Global Pages in the Data Translation Look-Aside Buffer (DTLB) May Not Be Flushed by RSM instruction before Restoring the Architectural State from SMRAM

Problem: The Resume from System Management Mode (RSM) instruction does not flush global pages from the Data Translation Look-Aside Buffer (DTLB) prior to reloading the saved architectural state.

Implication: If SMM turns on paging with global paging enabled and then maps any of linear addresses of SMRAM using global pages, RSM load may load data from the wrong location.

Workaround: Do not use global pages in system management mode.

Status: For the steppings affected, see the Summary Tables of Changes.

AH22. Sequential Code Fetch to Non-canonical Address May Have Nondeterministic Results

Problem: If code sequentially executes off the end of the positive canonical address space (falling through from address 00007fffffffffff to non- canonical address 0000800000000000), under some circumstances the code fetch will be converted to a canonical fetch at address ffff800000000000.

Implication: Due to this erratum, the processor may transfer control to an unintended address. The result of fetching code at that address is unpredictable and may include an unexpected trap or fault, or execution of the instructions found there.

Workaround: If the last page of the positive canonical address space is not allocated for code (4K page at 00007ffffffff000 or 2M page at 00007fffffe00000) then the problem cannot occur.

Status: For the steppings affected, see the Summary Tables of Changes.

AH23. VMCALL to Activate Dual-monitor Treatment of SMIs and SMM Ignores Reserved Bit Settings in VM-exit Control Field

Problem: Processors supporting Intel® Virtualization Technology can execute VMCALL from within the Virtual Machine Monitor (VMM) to activate dual-monitor treatment of SMIs and SMM. Due to this erratum, if reserved bits are set to values inconsistent with VMX Capability MSRs, VMCALL may not VMFail.

Implication: VMCALL executed to activate dual-monitor treatment of SMIs and SMM may not VMFail due to incorrect reserved bit settings in VM-Exit control field.

Workaround: Software should ensure that all VMCS reserved bits are set to values consistent with VMX Capability MSRs.

Status: For the steppings affected, see the Summary Tables of Changes.

AH24. REP MOVS/STOS Executing with Fast Strings Enabled and Crossing Page Boundaries with Inconsistent Memory Types May Use an Incorrect Data Size or Lead to Memory-Ordering Violations.

Problem: Under certain conditions as described in the Intel® 64 and IA-32 Architectures Software Developer's Manual, section Out-of-Order Stores For string operations in Pentium 4, Intel Xeon, and P6 Family Processors, the processor performs REP MOVS or REP STOS as fast strings. Due to this erratum fast string REP MOVS/REP STOS instructions that cross page boundaries from WB/WC memory types to UC/WP/WT memory types, may start using an incorrect data size or may observe memory ordering violations.

Implication: Upon crossing the page boundary the following may occur, dependent on the new page memory type:

1. UC the data size of each write will now always be 8 bytes, as opposed to the original data size.
2. WP the data size of each write will now always be 8 bytes, as opposed to the original data size and there may be a memory ordering violation.
3. WT there may be a memory ordering violation.

Workaround: Software should avoid crossing page boundaries from WB or WC memory type to UC, WP or WT memory type within a single REP MOVS or REP STOS instruction that will execute with fast strings enabled.

Status: For the steppings affected, see the Summary Tables of Changes.

AH25. Some Bus Performance Monitoring Events May Not Count Local Events under Certain Conditions

Problem: Many Performance Monitoring Events require core-specificity, which specifies which core's events are to be counted (local core, other core, or both cores). Due to this erratum, some Bus Performance Monitoring events may not count when the core-specificity is set to the local core.

Problem: The following Bus Transaction Performance Monitor events are supposed to count all local transactions:

- BUS_TRANS_ IO (Event: 6CH) – Will not count I/O level reads resulting from package resolved C-state
- BUS_TRANS_ANY (Event: 70H) – Will not count Stop-Grants

Implication: The count values for the affected events may be lower than expected. The degree of under count depends on the occurrence of erratum conditions while the affected events are active.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH26. Premature Execution of a Load Operation Prior to Exception Handler Invocation

Problem: If any of the below circumstances occur it is possible that the load portion of the instruction is executed before the exception handler is entered.

1. If an instruction that performs a memory load causes a code segment limit violation.
2. If a waiting X87 floating-point (FP) instruction or MMX™ technology instruction that performs a memory load has a floating-point exception pending.
3. If an MMX or SSE/SSE2/SSE3/SSSE3 extensions (SSE) instruction that performs a memory load and has either CR0.EM=1 (Emulation bit set), or a floating-point Top-of-Stack (FP TOS) not equal to 0, or a DNA exception pending.

Implication: In normal code execution where the target of the load operation is to write back memory there is no impact from the load being prematurely executed, or from the restart and subsequent re-execution of that instruction by the exception handler. If the target of the load is to uncached memory that has a system side-effect. Particularly, while CR0.TS [bit 3] is set, a MOVD/MOVQ with MMX/XMM register operands may issue a memory load before getting the DNA exception.

Workaround: Code which performs loads from memory that has side-effects can effectively workaround this behavior by using simple integer-based load instructions when accessing side-effect memory and by ensuring that all code is written such that a code segment limit violation cannot occur as a part of reading from side-effect memory.

Status: For the steppings affected, see the Summary Tables of Changes.

AH27. General Protection (#GP) Fault May Not Be Signaled on Data Segment Limit Violation above 4-G Limit

Problem: In 32-bit mode, memory accesses to flat data segments (base = 00000000h) that occur above the 4-G limit (0ffffffffh) may not signal a #GP fault.

Implication: When such memory accesses occur in 32-bit mode, the system may not issue a #GP fault.

Workaround: Software should ensure that memory accesses in 32-bit mode do not occur above the 4-G limit (0ffffffffh).

Status: For the steppings affected, see the Summary Tables of Changes.

AH28. EIP May Be Incorrect after Shutdown in IA-32e Mode

Problem: When the processor is going into shutdown state the upper 32 bits of the instruction pointer may be incorrect. This may be observed if the processor is taken out of shutdown state by NMI#.

Implication: A processor that has been taken out of the shutdown state may have an incorrect EIP. The only software which would be affected is diagnostic software that relies on a valid EIP.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH29. #GP Fault Is Not Generated on Writing IA32_MISC_ENABLE [34] When Execute Disable Is Not Supported

Problem: A #GP fault is not generated on writing to IA32_MISC_ENABLE [34] bit in a processor which does not support Execute Disable functionality.

Implication: Writing to IA32_MISC_ENABLE [34] bit is silently ignored without generating a fault.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH30. (E)CX May Get Incorrectly Updated When Performing Fast String REP MOVS or Fast String REP STOS with Large Data Structures

Problem: When performing Fast String REP MOVS or REP STOS commands with data structures [(E)CX*Data Size] larger than the supported address size structure (64 kB for 16-bit address size and 4 GB for 32-bit address size) some addresses may be processed more than once. After an amount of data greater than or equal to the address size structure has been processed, external events (such as interrupts) will cause the (E)CX registers to be increment by a value that corresponds to 64 kB for 16-bit address size and 4 GB for 32-bit address size.

Implication: (E)CX may contain an incorrect count which may cause some of the MOVS or STOS operations to re-execute. Intel has not observed this erratum with any commercially available software.

Workaround: Do not use values in (E)CX that when multiplied by the data size, give values larger than the address space size (64 kB for 16-bit address size and 4 GB for 32-bit address size).

Status: For the steppings affected, see the Summary Tables of Changes.

AH31. Performance Monitoring Events for Retired Loads (CBH) and Instructions Retired (C0H) May Not Be Accurate

Problem: The following events may be counted as instructions that contain a load by the MEM_LOAD_RETIRED performance monitor events and may be counted as loads by the INST_RETIRED (mask 01H) performance monitor event:

- Prefetch instructions
- x87 exceptions on FST* and FBSTP instructions
- Breakpoint matches on loads, stores, and I/O instructions
- Stores which update the A and D bits
- Stores that split across a cache line
- VMX transitions
- Any instruction fetch that misses in the ITLB

Implication: The MEM_LOAD_RETIRED and INST_RETIRED (mask 01H) performance monitor events may count a value higher than expected. The extent to which the values are higher than expected is determined by the frequency of the above events.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH32. Upper 32 Bits of 'From' Address Reported through BTMs or BTSs May Be Incorrect

Problem: When a far transfer switches the processor from 32-bit mode to IA-32e mode, the upper 32 bits of the 'From' (source) addresses reported through the BTMs (Branch Trace Messages) or BTSs (Branch Trace Stores) may be incorrect.

Implication: The upper 32 bits of the 'From' address debug information reported through BTMs or BTSs may be incorrect during this transition.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH33. Unsynchronized Cross-Modifying Code Operations Can Cause Unexpected Instruction Execution Results

Problem: The act of one processor, or system bus master, writing data into a currently executing code segment of a second processor with the intent of having the second processor execute that data as code is called cross-modifying code (XMC). XMC that does not force the second processor to execute a synchronizing instruction, prior to execution of the new code, is called unsynchronized XMC. Software using unsynchronized XMC to modify the instruction byte stream of a processor can see unexpected or unpredictable execution behavior from the processor that is executing the modified code.

Implication: In this case, the phrase "unexpected or unpredictable execution behavior" encompasses the generation of most of the exceptions listed in the *Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3: System Programming Guide*, including a General Protection Fault (GPF) or other unexpected behaviors. In the event that unpredictable execution causes a GPF the application executing the unsynchronized XMC operation would be terminated by the operating system.

Workaround: In order to avoid this erratum, programmers should use the XMC synchronization algorithm as detailed in the Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3: System Programming Guide, Section: Handling Self- and Cross-Modifying Code.

Status: Fixed. For the steppings affected, see the Summary Tables of Changes.

AH34. MSRs Actual Frequency Clock Count (IA32_APERF) or Maximum Frequency Clock Count (IA32_MPERF) May Contain Incorrect Data after a Machine Check Exception (MCE)

Problem: When an MCE occurs during execution of a RDMSR instruction for MSRs Actual Frequency Clock Count (IA32_APERF) or Maximum Frequency Clock Count (IA32_MPERF), the current and subsequent RDMSR instructions for these MSRs may contain incorrect data.

Implication: After an MCE event, accesses to the IA32_APERF and IA32_MPERF MSRs may return incorrect data. A subsequent reset will clear this condition.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH35. Incorrect Address Computed For Last Byte of FXSAVE/FXRSTOR Image Leads to Partial Memory Update

Problem: A partial memory state save of the 512-byte FXSAVE image or a partial memory state restore of the FXRSTOR image may occur if a memory address exceeds the 64KB limit while the processor is operating in 16-bit mode or if a memory address exceeds the 4GB limit while the processor is operating in 32-bit mode.

Implication: FXSAVE/FXRSTOR will incur a #GP fault due to the memory limit violation as expected but the memory state may be only partially saved or restored.

Workaround: Software should avoid memory accesses that wrap around the respective 16-bit and 32-bit mode memory limits.

Status: For the steppings affected, see the Summary Tables of Changes.

AH36. Split Locked Stores May Not Trigger the Monitoring Hardware

Problem: Logical processors normally resume program execution following the MWAIT, when another logical processor performs a write access to a WB cacheable address within the address range used to perform the MONITOR operation. Due to this erratum, a logical processor may not resume execution until the next targeted interrupt event or O/S timer tick following a locked store that spans across cache lines within the monitored address range.

Implication: The logical processor that executed the MWAIT instruction may not resume execution until the next targeted interrupt event or O/S timer tick in the case where the monitored address is written by a locked store which is split across cache lines.

Workaround: Do not use locked stores that span cache lines in the monitored address range.

Status: For the steppings affected, see the Summary Tables of Changes.

AH37. REP CMPS/SCAS Operations May Terminate Early in 64-bit Mode When RCX >= 0X100000000

Problem: REP CMPS (Compare String) and SCAS (Scan String) instructions in 64-bit mode may terminate before the count in RCX reaches zero if the initial value of RCX is greater than or equal to 0X100000000.

Implication: Early termination of REP CMPS/SCAS operation may be observed and RFLAGS may be incorrectly updated.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Tables of Changes.

AH38. FXSAVE/FXRSTOR Instructions which Store to the End of the Segment and Cause a Wrap to a Misaligned Base Address (Alignment <= 0x10h) May Cause FPU Instruction or Operand Pointer Corruption

Problem: If a FXSAVE/FXRSTOR instruction stores to the end of the segment causing a wrap to a misaligned base address (alignment <= 0x10h), and one of the following conditions is satisfied:

1. 32-bit addressing, obtained by using address-size override, when in 64-bit mode
2. 16-bit addressing in legacy or compatibility mode.

Then, depending on the wrap-around point, one of the below saved values may be corrupted:

- FPU Instruction Pointer Offset
- FPU Instruction Pointer Selector
- FPU Operand Pointer Selector
- FPU Operand Pointer Offset

Implication: This erratum could cause FPU Instruction or Operand pointer corruption and may lead to unexpected operations in the floating point exception handler.

Workaround: Avoid segment base misalignment and address wrap-around at the segment boundary.

Status: For the steppings affected, see the Summary Tables of Changes.

AH39. Cache Data Access Request from One Core Hitting a Modified Line in the L1 Data Cache of the Other Core May Cause Unpredictable System Behavior

Problem: When request for data from Core 1 results in a L1 cache miss, the request is sent to the L2 cache. If this request hits a modified line in the L1 data cache of Core 2, certain internal conditions may cause incorrect data to be returned to the Core 1.

Implication: This erratum may cause unpredictable system behavior.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Tables of Changes.

AH40. PREFETCHh Instruction Execution under Some Conditions May Lead to Processor Livelock

Problem: PREFETCHh instruction execution after a split load and dependent upon ongoing store operations may lead to processor livelock.

Implication: Due to this erratum, the processor may livelock.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Tables of Changes.

AH41. PREFETCHh Instructions May Not Be Executed When Alignment Check (AC) Is Enabled

Problem: PREFETCHT0, PREFETCHT1, PREFETCHT2 and PREFETCHNTA instructions may not be executed when Alignment Check is enabled.

Implication: PREFETCHh instructions may not perform the data prefetch if Alignment Check is enabled.

Workaround: Clear the AC flag (bit 18) in the EFLAGS register and/or the AM bit (bit 18) of Control Register CR0 to disable alignment checking.

Status: For the steppings affected, see the Summary Tables of Changes.

AH42. Upper 32 Bits of the FPU Data (Operand) Pointer in the FXSAVE Memory Image May Be Unexpectedly All 1's after FXSAVE

Problem: The upper 32 bits of the FPU Data (Operand) Pointer may incorrectly be set to all 1's instead of the expected value of all 0's in the FXSAVE memory image if all of the following conditions are true:

- The processor is in 64-bit mode.
- The last floating point operation was in compatibility mode
- Bit 31 of the FPU Data (Operand) Pointer is set.
- An FXSAVE instruction is executed

Implication: Software depending on the full FPU Data (Operand) Pointer may behave unpredictably.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH43. Concurrent Multi-processor Writes to Non-dirty Page May Result in Unpredictable Behavior

Problem: When a logical processor writes to a non-dirty page, and another logical-processor either writes to the same non-dirty page or explicitly sets the dirty bit in the corresponding page table entry, complex interaction with internal processor activity may cause unpredictable system behavior.

Implication: This erratum may result in unpredictable system behavior and hang.

Workaround: It is possible for BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Tables of Changes.

AH44. Performance Monitor IDLE_DURING_DIV (18h) Count May Not Be Accurate

Problem: Performance monitoring events that count the number of cycles the divider is busy and no other execution unit operation or load operation is in progress may not be accurate.

Implication: The counter may reflect a value higher or lower than the actual number of events.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH45. Values for LBR/BTS/BTM Will Be Incorrect after an Exit from SMM

Problem: After a return from SMM (System Management Mode), the CPU will incorrectly update the LBR (Last Branch Record) and the BTS (Branch Trace Store), hence rendering their data invalid. The corresponding data if sent out as a BTM on the system bus will also be incorrect.

Note: This issue would only occur when one of the 3 above mentioned debug support facilities are used.

Implication: The value of the LBR, BTS, and BTM immediately after an RSM operation should not be used.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH46. Shutdown Condition May Disable Non-Bootstrap Processors

Problem: When a logical processor encounters an error resulting in shutdown, non-bootstrap processors in the package may be unexpectedly disabled.

Implication: Non-bootstrap logical processors in the package that have not observed the error condition may be disabled and may not respond to INIT#, SMI#, NMI#, SIPI or other events.

Workaround: When this erratum occurs, RESET# must be asserted to restore multi-core functionality.

Status: For the steppings affected, see the Summary Tables of Changes.

AH47. SYSCALL Immediately after Changing EFLAGS.TF May Not Behave According to the New EFLAGS.TF

Problem: If a SYSCALL instruction follows immediately after EFLAGS.TF was updated and IA32_FMASK.TF (bit 8) is cleared, then under certain circumstances SYSCALL may behave according to the previous EFLAGS.TF.

Implication: When the problem occurs, SYSCALL may generate an unexpected debug exception, or may skip an expected debug exception.

Workaround: Mask EFLAGS.TF by setting IA32_FMASK.TF (bit 8).

Status: For the steppings affected, see the Summary Tables of Changes.

AH48. Code Segment Limit/Canonical Faults on RSM May Be Serviced before Higher Priority Interrupts/Exceptions and May Push the Wrong Address Onto the Stack

Problem: Normally, when the processor encounters a Segment Limit or Canonical Fault due to code execution, a #GP (General Protection Exception) fault is generated after all higher priority Interrupts and exceptions are serviced. Due to this erratum, if RSM (Resume from System Management Mode) returns to execution flow that results in a Code Segment Limit or Canonical Fault, the #GP fault may be serviced before a higher priority Interrupt or Exception (e.g. NMI (Non-Maskable Interrupt), Debug break(#DB), Machine Check (#MC), etc.). If the RSM attempts to return to a non-canonical address, the address pushed onto the stack for this #GP fault may not match the non-canonical address that caused the fault.

Implication: Operating systems may observe a #GP fault being serviced before higher priority Interrupts and Exceptions. Intel has not observed this erratum on any commercially available software.

Workaround: None Identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH49. VM Bit Is Cleared on Second Fault Handled by Task Switch from Virtual- 8086 (VM86)

Problem: Following a task switch to any fault handler that was initiated while the processor was in VM86 mode, if there is an additional fault while servicing the original task switch then the VM bit will be incorrectly cleared in EFLAGS, data segments will not be pushed and the processor will not return to the correct mode upon completion of the second fault handler via IRET.

Implication: When the OS recovers from the second fault handler, the processor will no longer be in VM86 mode. Normally, operating systems should prevent interrupt task switches from faulting, thus the scenario should not occur under normal circumstances.

Workaround: None Identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH50. IA32_FMASK Is Reset during an INIT

Problem: IA32_FMASK MSR (0xC0000084) is reset during INIT.

Implication: If an INIT takes place after IA32_FMASK is programmed, the processor will overwrite the value back to the default value.

Workaround: Operating system software should initialize IA32_FMASK after INIT.

Status: For the steppings affected, see the Summary Tables of Changes.

AH51. An Enabled Debug Breakpoint or Single Step Trap May Be Taken after MOV SS/POP SS Instruction if it is Followed by an Instruction That Signals a Floating Point Exception

Problem: A MOV SS/POP SS instruction should inhibit all interrupts including debug breakpoints until after execution of the following instruction. This is intended to allow the sequential execution of MOV SS/POP SS and MOV [r/e]SP, [r/e]BP instructions without having an invalid stack during interrupt handling. However, an enabled debug breakpoint or single step trap may be taken after MOV SS/POP SS if this instruction is followed by an instruction that signals a floating point exception rather than a MOV [r/e]SP, [r/e]BP instruction. This results in a debug exception being signaled on an unexpected instruction boundary since the MOV SS/POP SS and the following instruction should be executed atomically.

Implication: This can result in incorrect signaling of a debug exception and possibly a mismatched Stack Segment and Stack Pointer. If MOV SS/POP SS is not followed by a MOV [r/e]SP, [r/e]BP, there may be a mismatched Stack Segment and Stack Pointer on any exception. Intel has not observed this erratum with any commercially available software, or system.

Workaround: As recommended in the *Intel® 64 and IA-32 Architectures Software Developer's Manual*, the use of MOV SS/POP SS in conjunction with MOV [r/e]SP, [r/e]BP will avoid the failure since the MOV [r/e]SP, [r/e]BP will not generate a floating point exception. Developers of debug tools should be aware of the potential incorrect debug event signaling created by this erratum.

Status: For the steppings affected, see the Summary Tables of Changes.

AH52. Last Branch Records (LBR) Updates May Be Incorrect after a Task Switch

Problem: A Task-State Segment (TSS) task switch may incorrectly set the LBR_FROM value to the LBR_TO value.

Implication: The LBR_FROM will have the incorrect address of the Branch Instruction.

Workaround: None Identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH53. IO_SMI Indication in SMRAM State Save Area May Be Set Incorrectly

Problem: The IO_SMI bit in SMRAM's location 7FA4H is set to "1" by the CPU to indicate a System Management Interrupt (SMI) occurred as the result of executing an instruction that reads from an I/O port. Due to this erratum, the IO_SMI bit may be incorrectly set by:

- A non-I/O instruction.
- SMI is pending while a lower priority event interrupts.
- A REP I/O read.
- An I/O read that redirects to MWAIT.
- In systems supporting Intel® Virtualization Technology a fault in the middle of an IO operation that causes a VM Exit

Implication: SMM handlers may get false IO_SMI indication.

Workaround: The SMM handler has to evaluate the saved context to determine if the SMI was triggered by an instruction that read from an I/O port. The SMM handler must not restart an I/O instruction if the platform has not been configured to generate a synchronous SMI for the recorded I/O port address.

Status: For the steppings affected, see the Summary Tables of Changes.

AH54. INIT Does Not Clear Global Entries in the TLB

Problem: INIT may not flush a TLB entry when:

- The processor is in protected mode with paging enabled and the page global enable flag is set (PGE bit of CR4 register).
- G bit for the page table entry is set.
- TLB entry is present in TLB when INIT occurs.

Implication: Software may encounter unexpected page fault or incorrect address translation due to a TLB entry erroneously left in TLB after INIT.

Workaround: Write to CR3, CR4 (setting bits PSE, PGE or PAE) or CR0 (setting bits PG or PE) registers before writing to memory early in BIOS code to clear all the global entries from TLB.

Status: For the steppings affected, see the Summary Tables of Changes.

AH55. Using Memory Type Aliasing with Memory Types WB/WT May Lead to Unpredictable Behavior

Problem: Memory type aliasing occurs when a single physical page is mapped to two or more different linear addresses, each with different memory type. Memory type aliasing with the memory types WB and WT may cause the processor to perform incorrect operations leading to unpredictable behavior.

Implication: Software that uses aliasing of WB and WT memory types may observe unpredictable behavior.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Tables of Changes.

AH56. Update of Read/Write (R/W) or User/Supervisor (U/S) or Present (P) Bits without TLB Shootdown May Cause Unexpected Processor Behavior

Problem: Updating a page table entry by changing R/W, U/S or P bits without TLB shootdown (as defined by the 4 step procedure in "Propagation of Page Table and Page Directory Entry Changes to Multiple Processors" In volume 3A of the *Intel® 64 and IA-32 Architectures Software Developer's Manual*), in conjunction with a complex sequence of internal processor micro-architectural events, may lead to unexpected processor behavior.

Implication: This erratum may lead to livelock, shutdown or other unexpected processor behavior. Intel has not observed this erratum with any commercially available system.

Workaround: None Identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH57. BTS Message May Be Lost When the STPCLK# Signal Is Active

Problem: STPCLK# is asserted to enable the processor to enter a low-power state. Under some circumstances, when STPCLK# becomes active, the BTS (Branch Trace Store) message may be either lost and not written or written with corrupted branch address to the Debug Store area.

Implication: BTS messages may be lost or be corrupted in the presence of STPCLK# assertions.

Workaround: None Identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH58. MOV To/From Debug Registers Causes Debug Exception

Problem: When in V86 mode, if a MOV instruction is executed to/from a debug register, a general-protection exception (#GP) should be generated. However, in the case when the general detect enable flag (GD) bit is set, the observed behavior is that a debug exception (#DB) is generated instead.

Implication: With debug-register protection enabled (that is, the GD bit set), when attempting to execute a MOV on debug registers in V86 mode, a debug exception will be generated instead of the expected general-protection fault.

Workaround: In general, operating systems do not set the GD bit when they are in V86 mode. The GD bit is generally set and used by debuggers. The debug exception handler should check that the exception did not occur in V86 mode before continuing. If the exception did occur in V86 mode, the exception may be directed to the general-protection exception handler.

Status: For the steppings affected, see the Summary Tables of Changes.

AH59. EFLAGS Discrepancy on a Page Fault after a Multiprocessor TLB Shootdown

Problem: This erratum may occur when the processor executes one of the following read-modify-write arithmetic instructions and a page fault occurs during the store of the memory operand: ADD, AND, BTC, BTR, BTS, CMPXCHG, DEC, INC, NEG, NOT, OR, ROL/ROR, SAL/SAR/SHL/SHR, SHLD, SHRD, SUB, XOR, and XADD. In this case, the EFLAGS value pushed onto the stack of the page fault handler may reflect the status of the register after the instruction would have completed execution rather than before it. The following conditions are required for the store to generate a page fault and call the operating system page fault handler:

1. The store address entry must be evicted from the DTLB by speculative loads from other instructions that hit the same way of the DTLB before the store has completed. DTLB eviction requires at least three-load operations that have linear address bits 15:12 equal to each other and address bits 31:16 different from each other in close physical proximity to the arithmetic operation.
2. The page table entry for the store address must have its permissions tightened during the very small window of time between the DTLB eviction and execution of the store. Examples of page permission tightening include from Present to Not Present or from Read/Write to Read Only, etc. 3. Another processor, without corresponding synchronization and TLB flush, must cause the permission change.

Implication: This scenario may only occur on a multiprocessor platform running an operating system that performs "lazy" TLB shootdowns. The memory image of the EFLAGS register on the page fault handler's stack prematurely contains the final arithmetic flag values although the instruction has not yet completed. Intel has not identified any operating systems that inspect the arithmetic portion of the EFLAGS register during a page fault nor observed this erratum in laboratory testing of software applications.

Workaround: No workaround is needed upon normal restart of the instruction, since this erratum is transparent to the faulting code and results in correct instruction behavior. Operating systems may ensure that no processor is currently accessing a page that is scheduled to have its page permissions tightened or have a page fault handler that ignores any incorrect state.

Status: For the steppings affected, see the Summary Tables of Changes.

AH60. LBR, BTS, BTM May Report a Wrong Address When an Exception/Interrupt Occurs in 64-bit Mode

Problem: An exception/interrupt event should be transparent to the LBR (Last Branch Record), BTS (Branch Trace Store) and BTM (Branch Trace Message) mechanisms. However, during a specific boundary condition where the exception/interrupt occurs right after the execution of an instruction at the lower canonical boundary (0x00007FFFFFFFFFFF) in 64-bit mode, the LBR return registers will save a wrong return address with bits 63 to 48 incorrectly sign extended to all 1's. Subsequent BTS and BTM operations which report the LBR will also be incorrect.

Implication: LBR, BTS and BTM may report incorrect information in the event of an exception/interrupt.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH61. A Thermal Interrupt Is Not Generated When the Current Temperature Is Invalid

Problem: When the DTS (Digital Thermal Sensor) crosses one of its programmed thresholds it generates an interrupt and logs the event (IA32_THERM_STATUS MSR (019Ch) bits [9, 7]). Due to this erratum, if the DTS reaches an invalid temperature (as indicated IA32_THERM_STATUS MSR bit[31]) it does not generate an interrupt even if one of the programmed thresholds is crossed and the corresponding log bits become set.

Implication: When the temperature reaches an invalid temperature the CPU does not generate a Thermal interrupt even if a programmed threshold is crossed.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH62. CMPSB, LODSB, or SCASB in 64-bit Mode with Count Greater or Equal to 2^{48} May Terminate Early

Problem: In 64-bit Mode CMPSB, LODSB, or SCASB executed with a repeat prefix and count greater than or equal to 2^{48} may terminate early. Early termination may result in one of the following.

- The last iteration not being executed
- Signaling of a canonical limit fault (#GP) on the last iteration

Implication: While in 64-bit mode, with count greater or equal to 2^{48}, repeat string operations CMPSB, LODSB or SCASB may terminate without completing the last iteration. Intel has not observed this erratum with any commercially available software.

Workaround: Do not use repeated string operations with RCX greater than or equal to 2^{48}.

Status: For the steppings affected, see the Summary Tables of Changes.

AH63. Removed Erratum

AH64. Returning to Real Mode from SMM with EFLAGS.VM Set May Result in Unpredictable System Behavior

Problem: Returning back from SMM mode into real mode while EFLAGS.VM is set in SMRAM may result in unpredictable system behavior.

Implication: If SMM software changes the values of the EFLAGS.VM in SMRAM, it may result in unpredictable system behavior. Intel has not observed this behavior in commercially available software.

Workaround: SMM software should not change the value of EFLAGS.VM in SMRAM.

Status: For the steppings affected, see the Summary Tables of Changes.

AH65. VMLAUNCH/VMRESUME May Not Fail when VMCS Is Programmed to Cause VM Exit to Return to a Different Mode

Problem: VMLAUNCH/VMRESUME instructions may not fail if the value of the "host address-space size" VM-exit control differs from the setting of IA32_EFER.LMA.

Implication: Programming the VMCS to allow the monitor to be in different modes prior to VMLAUNCH/VMRESUME and after VM-exit may result in undefined behavior

Workaround: Software should ensure that "host address-space size" VM-exit control has the same value as IA32_EFER.LMA at the time of VMLAUNCH/VMRESUME.

Status: For the steppings affected, see the Summary Tables of Changes.

AH66. IRET under Certain Conditions May Cause an Unexpected Alignment Check Exception

Problem: In IA-32e mode, it is possible to get an Alignment Check Exception (#AC) on the IRET instruction even though alignment checks were disabled at the start of the IRET. This can only occur if the IRET instruction is returning from CPL3 code to CPL3 code. IRETs from CPL0/1/2 are not affected. This erratum can occur if the EFLAGS value on the stack has the AC flag set, and the interrupt handler's stack is misaligned. In IA-32e mode, RSP is aligned to a 16-byte boundary before pushing the stack frame.

Implication: In IA-32e mode, under the conditions given above, an IRET can get a #AC even if alignment checks are disabled at the start of the IRET. This erratum can only be observed with a software generated stack frame.

Workaround: Software should not generate misaligned stack frames for use with IRET.

Status: For the steppings affected, see the Summary Tables of Changes.

AH67. Performance Monitoring Event FP_ASSIST May Not Be Accurate

Problem: Performance monitoring event FP_ASSIST (11H) may be inaccurate as assist events will be counted twice per actual assist in the following specific cases:

- FADD and FMUL instructions with a NaN(Not a Number) operand and a memory operand
- FDIV instruction with zero operand value in memory

Problem: In addition, an assist event may be counted when DAZ (Denormals-Are-Zeros) and FTZ (Flush-To-Zero) flags are turned on even though no actual assist occurs.

Implication: The counter value for the performance monitoring event FP_ASSIST (11H) may be larger than expected. The size of the error is dependent on the number of occurrences of the above conditions while the event is active.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH68. CPL-Qualified BTS May Report Incorrect Branch-From Instruction Address

Problem: CPL (Current Privilege Level)-qualified BTS (Branch Trace Store) may report incorrect branch-from instruction address under the following conditions:

- Either BTS_OFF_OS [9] or BTS_OFF_USR [10] is selected in IA32_DEBUGCTLC MSR (1D9H).
- Privilege-level transitions occur between CPL > 0 and CPL 0 or vice versa.

Implication: Due to this erratum, the From address reported by BTS may be incorrect for the described conditions.

Workaround: None Identified

Status: For the steppings affected, see the Summary Tables of Changes.

AH69. PEBS Does Not Always Differentiate Between CPL-Qualified Events

Problem: Performance monitoring counter configured to sample PEBS (Precise Event Based Sampling) events at a certain privilege level may count samples at the wrong privilege level.

Implication: Performance monitoring counter may be higher than expected for CPL-qualified events.

Workaround: Do not use performance monitoring counters for precise event sampling when the precise event is dependent on the CPL value.

Status: For the steppings affected, see the Summary Tables of Changes.

AH70. PMI May Be Delayed to Next PEBS Event

Problem: After a PEBS (Precise Event-Based Sampling) event, the PEBS index is compared with the PEBS threshold, and the index is incremented with every event. If PEBS index is equal to the PEBS threshold, a PMI (Performance Monitoring Interrupt) should be issued. Due to this erratum, the PMI may be delayed by one PEBS event.

Implication: Debug Store Interrupt Service Routines may observe delay of PMI occurrence by one PEBS event.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH71. PEBS Buffer Overflow Status Will Not Be Indicated Unless IA32_DEBUGCTL[12] Is Set

Problem: IA32_PERF_GLOBAL_STATUS MSR (38EH) bit [62] when set, indicates that a PEBS (Precise Event-Based Sampling) overflow has occurred and a PMI (Performance Monitor Interrupt) has been sent. Due to this erratum, this bit is not set unless IA32_DEBUGCTL MSR (1D9H) bit [12] (which stops all performance monitor counters upon a PMI) is also set.

Implication: Due to this erratum, IA32_PERF_GLOBAL_STATUS [62] will not signal that a PMI was generated due to a PEBS Overflow unless IA32_DEBUGCTL [12] is set.

Workaround: It is possible for the software to set IA32_DEBUGCTL [12] to avoid this erratum.

Status: For the steppings affected, see the Summary Tables of Changes.

AH72. The BS Flag in DR6 May Be Set for Non-Single-Step #DB Exception

Problem: DR6 BS (Single Step, bit 14) flag may be incorrectly set when the TF (Trap Flag, bit 8) of the EFLAGS Register is set, and a #DB (Debug Exception) occurs due to one of the following:

- DR7 GD (General Detect, bit 13) being bit set
- INT1 instruction;
- Code breakpoint

Implication: The BS flag may be incorrectly set for non-single-step #DB exception.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH73. An Asynchronous MCE During a Far Transfer May Corrupt ESP

Problem: If an asynchronous machine check occurs during an interrupt, call through gate, FAR RET or IRET and in the presence of certain internal conditions, ESP may be corrupted.

Implication: If the MCE (Machine Check Exception) handler is called without a stack switch, then a triple fault will occur due to the corrupted stack pointer, resulting in a processor shutdown. If the MCE is called with a stack switch, for example when the CPL (Current Privilege Level) was changed or when going through an interrupt task gate, then the corrupted ESP will be saved on the stack or in the TSS (Task State Segment), and will not be used.

Workaround: Use an interrupt task gate for the machine check handler.

Status: For the steppings affected, see the Summary Tables of Changes.

AH74 In Single-Stepping on Branches Mode, the BS Bit in the Pending-Debug-Exceptions Field of the Guest State Area Will Be Incorrectly Set by VM-Exit on a MOV to CR8 Instruction

Problem: In a system supporting Intel Virtualization Technology, the BS bit (bit 14 of the Pending-Debug-Exceptions field) in the guest state area will be incorrectly set when all of the following conditions occur:

The processor is running in VMX non-root as a 64 bit mode guest;

- The "CR8-load existing" VM-execution control is 0 and the "use TPR shadow" VMexecution is 1.
- Both BTF (Single-Step On Branches, bit 1) of the IA32_DEBUGCTL MSR (1D9H) Register and the TF (Trap Flag, bit 8) of the RFLAGS Register are set.
- "MOV CR8, reg" attempts to program a TPR (Task Priority Register) value that is below the TPR threshold and causes a VM-exit.

Implication: A Virtual-Machine will sample the BS bit and will incorrectly inject a Single-Step trap to the guest.

Workaround: A Virtual-Machine Monitor must manually disregard the BS bit in the Guest State Area in case of a VM-exit due to a TPR value below the TPR threshold.

Status: For the steppings affected, see the Summary Tables of Changes.

AH75. B0-B3 Bits in DR6 May Not Be Properly Cleared after Code Breakpoint

Problem: B0-B3 bits (breakpoint conditions detect flags, bits [3:0]) in DR6 may not be properly cleared when the following sequence happens:

1. POP instruction to SS (Stack Segment) selector.
2. Next instruction is FP (Floating Point) that gets FP assist followed by code breakpoint.

Implication: B0-B3 bits in DR6 may not be properly cleared.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH76. BTM/BTS Branch-From Instruction Address May Be Incorrect for Software Interrupts

Problem: When BTM (Branch Trace Message) or BTS (Branch Trace Store) is enabled, a software interrupt may result in the overwriting of BTM/BTS branch-from instruction address by the LBR (Last Branch Record) branch-from instruction address.

Implication: A BTM/BTS branch-from instruction address may get corrupted for software interrupts.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH77. REP Store Instructions in a Specific Situation May Cause the Processor to Hang

Problem: During a series of REP (repeat) store instructions a store may try to dispatch to memory prior to the actual completion of the instruction. This behavior depends on the execution order of the instructions, the timing of a speculative jump and the timing of an uncacheable memory store. All types of REP store instructions are affected by this erratum.

Implication: When this erratum occurs, the processor may live lock and/or result in a system hang.

Workaround: It is possible for BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Tables of Changes.

AH78. Performance Monitor SSE Retired Instructions May Return Incorrect Values

Problem: The SIMD_INST_RETIRED (Event: C7H) is used to track retired SSE instructions. Due to this erratum, the processor may also count other types of instructions resulting in values higher than the number of actual retired SSE instructions.

Implication: The event monitor instruction SIMD_INST_RETIRED may report count higher than expected.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH79. Performance Monitoring Events for L1 and L2 Miss May Not Be Accurate

Problem: Performance monitoring events 0CBh with an event mask value of 02h or 08h (MEM_LOAD_RETIRED.L1_LINE_MISS or MEM_LOAD_RETIRED.L2_LINE_MISS) may under count the cache miss events.

Implication: These performance monitoring events may show a count which is lower than expected; the amount by which the count is lower is dependent on other conditions occurring on the same load that missed the cache.

Workaround: None Identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH80. Store to WT Memory Data May Be Seen in Wrong Order by Two Subsequent Loads

Problem: When data of Store to WT memory is used by two subsequent loads of one thread and another thread performs cacheable write to the same address the first load may get the data from external memory or L2 written by another core, while the second load will get the data straight from the WT Store.

Implication: Software that uses WB to WT memory aliasing may violate proper store ordering.

Workaround: Do not use WB to WT aliasing.

Status: For the steppings affected, see the Summary Tables of Changes.

AH81. A MOV Instruction from CR8 Register with 16-Bit Operand Size Will Leave Bits 63:16 of the Destination Register Unmodified

Problem: Moves to/from control registers are supposed to ignore REW.W and the 66H (operand size) prefix. In systems supporting Intel Virtualization Technology, when the processor is operating in VMX non-root operation and "use TPR shadow" VM-execution control is set to 1, a MOV instruction from CR8 with a 16 bit operand size (REX.W =0 and 66H prefix) will only store 16 bits and leave bits 63:16 at the destination register unmodified, instead of storing zeros in them.

Implication: Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH82. Debug Register May Contain Incorrect Information on a MOVSS or POPSS Instruction Followed by SYSRET

Problem: In IA-32e mode, if a MOVSS or POPSS instruction with a debug breakpoint is followed by the SYSRET instruction; incorrect information may exist in the Debug Status Register (DR6).

Implication: When debugging or when developing debuggers, this behavior should be noted. This erratum does not occur under normal usage of the MOVSS or POPSS instructions (that is, following them with a MOV ESP instruction).

Workaround: Do not attempt to put a breakpoint on MOVSS and POPSS instructions that are followed by a SYSRET.

Status: For the steppings affected, see the Summary Tables of Changes.

AH83. Single Step Interrupts with Floating Point Exception Pending May Be Mishandled

Problem: In certain circumstances, when a floating point exception (#MF) is pending during single-step execution, processing of the single-step debug exception (#DB) may be mishandled.

Implication: When this erratum occurs, #DB will be incorrectly handled as follows:

- #DB is signaled before the pending higher priority #MF (Interrupt 16)
- #DB is generated twice on the same instruction

Workaround: None Identified

Status: For the steppings affected, see the Summary Tables of Changes.

AH84. Non-Temporal Data Store May Be Observed in Wrong Program Order

Problem: When non-temporal data is accessed by multiple read operations in one thread while another thread performs a cacheable write operation to the same address, the data stored may be observed in wrong program order (i.e., later load operations may read older data).

Implication: Software that uses non-temporal data without proper serialization before accessing the non-temporal data may observe data in wrong program order.

Workaround: Software that conforms to the *Intel® 64 and IA-32 Architecture Software Developer's Manual, Volume 3A*, section "Buffering of Write Combining Memory Locations" will operate correctly.

Status: For the steppings affected, see the Summary Tables of Changes.

AH85. Fault on ENTER Instruction May Result in Unexpected Values on Stack Frame

Problem: The ENTER instruction is used to create a procedure stack frame. Due to this erratum, if execution of the ENTER instruction results in a fault, the dynamic storage area of the resultant stack frame may contain unexpected values (i.e., residual stack data as a result of processing the fault).

Implication: Data in the created stack frame may be altered following a fault on the ENTER instruction. Please refer to "Procedure Calls For Block-Structured Languages" in the *Intel® 64 and IA-32 Architectures Software Developer's Manual, Vol. 1, Basic Architecture*, for information on the usage of the ENTER instructions. This erratum is not expected to occur in ring 3. Faults are usually processed in ring 0 and stack switch occurs when transferring to ring 0. Intel has not observed this erratum on any commercially-available software.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH86. CPUID Reports Architectural Performance Monitoring Version 2 Is Supported, When Only Version 1 Capabilities Are Available

Problem: CPUID leaf 0Ah reports the architectural performance monitoring version that is available in EAX[7:0]. Due to this erratum CPUID reports the supported version as 2 instead of 1.

Implication: Software will observe an incorrect version number in CPUID.0Ah.EAX [7:0] in comparison to which features are actually supported.

Workaround: Software should use the recommended enumeration mechanism described in the Architectural Performance Monitoring section of the *Intel® 64 and IA-32 Architecture Software Developer's Manual, Volume 3: System Programming Guide.*

Status: For the steppings affected, see the Summary Tables of Changes.

AH87. Unaligned Accesses to Paging Structures May Cause the Processor to Hang

Problem: When an unaligned access is performed on paging structure entries, accessing a portion of two different entries simultaneously, the processor may live lock.

Implication: When this erratum occurs, the processor may live lock causing a system hang.

Workaround: Do not perform unaligned accesses on paging structure entries.

Status: For the steppings affected, see the Summary Tables of Changes.

AH88. Microcode Updates Performed During VMX Non-root Operation Could Result in Unexpected Behavior

Problem: When Intel® Virtualization Technology is enabled, microcode updates are allowed only during VMX root operations. Attempts to apply microcode updates while in VMX non-root operation should be silently ignored. Due to this erratum, the processor may allow microcode updates during VMX non-root operations if not explicitly prevented by the host software.

Implication: Microcode updates performed in non-root operation may result in unexpected system behavior.

Workaround: Host software should intercept and prevent loads to IA32_BIOS_UPDT_TRIG MSR (79H) during VMX non-root operations. There are two mechanism that can be used (1) Enabling MSR access protection in the VM-execution controls or (2) Enabling selective MSR protection of IA32_BIOS_UPDT_TRIG MSR.

Status: For the steppings affected, see the Summary Tables of Changes.

AH89. INVLPG Operation for Large (2M/4M) Pages May Be Incomplete under Certain Conditions

Problem: The INVLPG instruction may not completely invalidate Translation Look-aside Buffer (TLB) entries for large pages (2-M/4-M) when both of the following conditions exist:

- Address range of the page being invalidated spans several Memory Type Range Registers (MTRRs) with different memory types specified.
- INVLPG operation is preceded by a Page Assist Event (Page Fault (#PF) or an access that results in either A or D bits being set in a Page Table Entry (PTE)

Implication: Stale translations may remain valid in TLB after a PTE update resulting in unpredictable system behavior. Intel has not observed this erratum with any commercially available software

Workaround: Software should ensure that the memory type specified in the MTRRs is the same for the entire address range of the large page.

Status: For the steppings affected, see the Summary Tables of Changes.

AH90. Page Access Bit May Be Set Prior to Signaling a Code Segment Limit Fault

Problem: If code segment limit is set close to the end of a code page, then due to this erratum the memory page Access bit (A bit) may be set for the subsequent page prior to general protection fault on code segment limit.

Implication: When this erratum occurs, a non-accessed page present in memory following a page that contains the code segment limit may be tagged as accessed

Workaround: Non-present or non-executable page can be placed after the limit of the code segment to prevent this erratum.

Status: For the steppings affected, see the Summary Tables of Changes.

AH91. Update of Attribute Bits on Page Directories without Immediate TLB Shootdown May Cause Unexpected Processor Behavior

Problem: Updating a page directory entry (or page map level 4 table entry or page directory pointer table entry in IA-32e mode) by changing R/W, U/S or P bits without immediate TLB shootdown (as described by the 4 step procedure in "Propagation of Page Table and Page Directory Entry Changes to Multiple Processors" In Volume 3A of the *Intel® 64 and IA-32 Architectures Software Developer's Manual*), in conjunction with a complex sequence of internal processor micro-architectural events, may lead to unexpected processor behavior.

Implication: This erratum may lead to livelock, shutdown or other unexpected processor behavior. Intel has not observed this erratum with any commercially available system.

Workaround: It is possible for BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Tables of Changes.

AH92. Invalid Instructions May Lead to Unexpected Behavior

Problem: Invalid instructions due to undefined opcodes or instructions exceeding the maximum instruction length (due to redundant prefixes placed before the instruction) may lead, under complex circumstances, to unexpected behavior.

Implication: The processor may behave unexpectedly due to invalid instructions. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH93. EFLAGS, CR0, CR4 and the EXF4 Signal May Be Incorrect after Shutdown

Problem: When the processor is going into shutdown due to an RSM inconsistency failure, EFLAGS, CR0 and CR4 may be incorrect. In addition the EXF4 signal may still be asserted. This may be observed if the processor is taken out of shutdown by NMI#.

Implication: A processor that has been taken out of shutdown may have an incorrect EFLAGS, CR0 and CR4. In addition the EXF4 signal may still be asserted.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH94. Performance Monitoring Counter MACRO_INSTS.DECODED May Not Count Some Decoded Instructions

Problem: MACRO_INSTS.DECODED performance monitoring counter (Event 0AAH, Umask 01H) counts the number of macro instructions decoded, but not necessarily retired. The event is undercounted when the decoded instructions are a complete loop iteration that is decoded in one cycle and the loop is streamed by the LSD (Loop Stream Detector), as described in the Optimizing the Front End section of the *Intel® 64 and IA-32 Architectures Optimization Reference Manual.*

Implication: The count value returned by the performance monitoring counter MACRO_INST.DECODED may be lower than expected. The degree of undercounting is dependent on the occurrence of loop iterations that are decoded in one cycle and whether the loop is streamed by the LSD while the counter is active.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH95. The Stack May Be Incorrect as a Result of VIP/VIF Check on SYSEXIT and SYSRET

Problem: The stack size may be incorrect under the following scenario:

Problem: 1. The stack size was changed due to a SYSEXIT or SYSRET

Problem: 2. PVI (Protected Mode Virtual Interrupts) mode was enabled (CR4.PVI == 1)

Problem: 3. Both the VIF (Virtual Interrupt Flag) and VIP (Virtual Interrupt Pending) flags of the EFLAGS register are set

Implication: If this erratum occurs the stack size may be incorrect, consequently this may result in unpredictable system behavior. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH96. Performance Monitoring Event SIMD_UOP_TYPE_EXEC.MUL Is Counted Incorrectly for PMULUDQ Instruction

Problem: Performance Monitoring Event SIMD_UOP_TYPE_EXEC.MUL (Event select 0B3H, Umask 01H) counts the number of SIMD packed multiply micro-ops executed. The count for PMULUDQ micro-ops might be lower than expected. No other instruction is affected.

Implication: The count value returned by the performance monitoring event SIMD_UOP_TYPE_EXEC.MUL may be lower than expected. The degree of undercount depends on actual occurrences of PMULUDQ instructions, while the counter is active.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH97. Storage of PEBS Record Delayed Following Execution of MOV SS or STI

Problem: When a performance monitoring counter is configured for PEBS (Precise Event Based Sampling), overflow of the counter results in storage of a PEBS record in the PEBS buffer. The information in the PEBS record represents the state of the next instruction to be executed following the counter overflow. Due to this erratum, if the counter overflow occurs after execution of either MOV SS or STI, storage of the PEBS record is delayed by one instruction.

Implication: When this erratum occurs, software may observe storage of the PEBS record being delayed by one instruction following execution of MOV SS or STI. The state information in the PEBS record will also reflect the one instruction delay.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH98. Updating Code Page Directory Attributes without TLB Invalidation May Result in Improper Handling of Code #PF

Problem: Code #PF (Page Fault exception) is normally handled in lower priority order relative to both code #DB (Debug Exception) and code Segment Limit Violation #GP (General Protection Fault). Due to this erratum, code #PF may be handled incorrectly, if all of the following conditions are met:

Problem: A PDE (Page Directory Entry) is modified without invalidating the corresponding TLB (Translation Look-aside Buffer) entry

Problem: Code execution transitions to a different code page such that both
- The target linear address corresponds to the modified PDE
- The PTE (Page Table Entry) for the target linear address has an A (Accessed) bit that is clear

Problem: One of the following simultaneous exception conditions is present following the code transition
- Code #DB and code #PF
- Code Segment Limit Violation #GP and code #PF

Implication: Software may observe either incorrect processing of code #PF before code Segment Limit Violation #GP or processing of code #PF in lieu of code #DB.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH99. Performance Monitoring Event CPU_CLK_UNHALTED.REF May Not Count Clock Cycles According to the Processors Operating Frequency

Problem: Performance Counter MSR_PERF_FIXED_CTR2 (MSR 30BH) that counts CPU_CLK_UNHALTED.REF clocks should count these clock cycles at a constant rate that is determined by the maximum resolved boot frequency, as programmed by BIOS. Due to this erratum, the rate is instead set by the maximum core-clock to bus-clock ratio of the processor, as indicated by hardware.

Implication: No functional impact as a result of this erratum. If the maximum resolved boot frequency as programmed by BIOS is different from the frequency implied by the maximum core-clock to bus-clock ratio of the processor as indicated by hardware, then the following effects may be observed:

Workaround: Performance Monitoring Event CPU_CLK_UNHALTED.REF will count at a rate different than the TSC (Time Stamp Counter)

- When running a system with several processors that have different maximum core-clock to bus-clock ratios, CPU_CLK_UNHALTED.REF monitoring events at each processor will be counted at different rates and therefore will not be comparable.
- Calculate the ratio of the rates at which the TSC and the CPU_CLK_UNHALTED.REF performance monitoring event count (this can be done by measuring simultaneously their counted value while executing code) and adjust the CPU_CLK_UNHALTED.REF event count to the maximum resolved boot frequency using this ratio.

Status: For the steppings affected, see the Summary Tables of Changes.

AH100. Store Ordering May Be Incorrect between WC and WP Memory Types

Problem: According to *Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3A, Methods of Caching Available*, WP (Write Protected) stores should drain the WC (Write Combining) buffers in the same way as UC (Uncacheable) memory type stores do. Due to this erratum, WP stores may not drain the WC buffers.

Implication: Memory ordering may be violated between WC and WP stores.

Workaround: None Identified

Status: For the steppings affected, see the Summary Tables of Changes.

AH101. (E)CX May Get Incorrectly Updated When Performing Fast String REP STOS with Large Data Structures

Problem: When performing Fast String REP STOS commands with data structures [(E)CX*Data Size] larger than the supported address size structure (64K for 16-bit address size and 4G for 32-bit address size) some addresses may be processed more than once. After an amount of data greater than or equal to the address size structure has been processed, external events (such as interrupts) will cause the (E)CX registers to be incremented by a value that corresponds to 64K bytes for 16 bit address size and 4G bytes for 32 bit address size.

Implication: (E)CX may contain an incorrect count which may cause some of the STOS operations to re-execute. Intel has not observed this erratum with any commercially available software.

Workaround: Do not use values in (E)CX that when multiplied by the data size, give values larger than the address space size (64K for 16-bit address size and 4G for 32-bit address size).

Status: For the steppings affected, see the Summary Tables of Changes.

AH102. Performance Monitoring Event BR_INST_RETIRED May Count CPUID Instructions as Branches

Problem: Performance monitoring event BR_INST_RETIRED (C4H) counts retired branch instructions. Due to this erratum, two of its sub-events mistakenly count for CPUID instructions as well. Those sub events are: BR_INST_RETIRED.PRED_NOT_TAKEN (Umask 01H) and BR_INST_RETIRED.ANY (Umask 00H).

Implication: The count value returned by the performance monitoring event BR_INST_RETIRED.PRED_NOT_TAKEN or BR_INST_RETIRED.ANY may be higher than expected. The extent of over counting depends on the occurrence of CPUID instructions, while the counter is active.

Workaround: None Identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH103. Performance Monitoring Event MISALIGN_MEM_REF May Over Count

Problem: Performance monitoring event MISALIGN_MEM_REF (05H) is used to count the number of memory accesses that cross an 8-byte boundary and are blocked until retirement. Due to this erratum, the performance monitoring event MISALIGN_MEM_REF also counts other memory accesses.

Implication: The performance monitoring event MISALIGN_MEM_REF may over count. The extent of the over counting depends on the number of memory accesses retiring while the counter is active.

Workaround: None Identified

Status: For the steppings affected, see the Summary Tables of Changes.

AH104. A REP STOS/MOVS to a MONITOR/MWAIT Address Range May Prevent Triggering of the Monitoring Hardware

Problem: The MONITOR instruction is used to arm the address monitoring hardware for the subsequent MWAIT instruction. The hardware is triggered on subsequent memory store operations to the monitored address range. Due to this erratum, REP STOS/MOVS fast string operations to the monitored address range may prevent the actual triggering store to be propagated to the monitoring hardware.

Implication: A logical processor executing an MWAIT instruction may not immediately continue program execution if a REP STOS/MOVS targets the monitored address range.

Workaround: Software can avoid this erratum by not using REP STOS/MOVS store operations within the monitored address range.

Status: For the steppings affected, see the Summary Tables of Changes.

AH105. False Level One Data Cache Parity Machine-Check Exceptions May Be Signaled

Problem: Executing an instruction stream containing invalid instructions/data may generate a false Level One Data Cache parity machine-check exception.

Implication: The false Level One Data Cache parity machine-check exception is reported as an uncorrected machine-check error. An uncorrected machine-check error is treated as a fatal exception by the operating system and may cause a shutdown and/or reboot.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Tables of Changes.

AH106. A Memory Access May Get a Wrong Memory Type Following a #GP due to WRMSR to an MTRR Mask

Problem: The TLB (Translation Lookaside Buffer) may indicate a wrong memory type on a memory access to a large page (2M/4M Byte) following the recovery from a #GP (General Protection Fault) due to a WRMSR to one of the IA32_MTRR_PHYSMASKn MSRs with reserved bits set.

Implication: When this erratum occurs, a memory access may get an incorrect memory type leading to unexpected system operation. As an example, an access to a memory mapped I/O device may be incorrectly marked as cacheable, become cached, and never make it to the I/O device. Intel has not observed this erratum with any commercially available software.

Workaround: Software should not attempt to set reserved bits of IA32_MTRR_PHYSMASK*n* MSRs.

Status: For the steppings affected, see the Summary Tables of Changes.

AH107. PMI While LBR Freeze Enabled May Result in Old/Out-of-Date LBR Information

Problem: When Precise Event-Based Sampling (PEBS) is configured with Performance Monitoring Interrupt (PMI) on PEBS buffer overflow enabled and Last Branch Record (LBR) Freeze on PMI enabled by setting FREEZE_LBRS_ON_PMI flag (bit 11) to 1 in IA32_DEBUGCTL (MSR 1D9H), the LBR stack is frozen upon the occurrence of a hardware PMI request. Due to this erratum, the LBR freeze may occur too soon (i.e. before the hardware PMI request).

Implication: Following a PMI occurrence, the PMI handler may observe old/out-of-date LBR information that does not describe the last few branches before the PEBS sample that triggered the PMI.

Workaround: None identified

Status: For the steppings affected, see the Summary Tables of Changes.

AH108. Overlap of an Intel® VT APIC Access Page in a Guest with the DS Save Area May Lead to Unpredictable Behavior

Problem: Logging of a branch record or a PEBS (precise-event-based-sampling) record to the DS (debug store) save area that overlaps with the APIC access page may lead to unpredictable behavior.

Implication: Guest software configured to log branch records or PEBS records cannot specify the DS (debug store) save area within the APIC-access page. Under any expected usage model this type of overlap is not expected to exist. One should be aware of the fact that the specified DS address is of linear form while the APIC access page is of a physical form. Any solution that wishes to avoid this condition will need to comprehend the linear-to-physical translation of the DS related address pointers with respect to the mapping of the physical APIC access page to avoid such an overlap. Under normal circumstances for correctly written software, such an overlap is not expected to exist. Intel has not observed this erratum with any commercially available software.

Workaround: For a fully comprehensive workaround, the VMM should not allow the logging of branch or PEBS records while guest software is running if the "virtualize APIC accesses" VM-execution control is 1.

Status: For the steppings affected, see the Summary Tables of Changes.

AH109. VTPR Write Access during Event Delivery May Cause an APIC-Access VM Exit

Problem: VTPR write accesses should not cause APIC-access VM exits but instead should cause data to be written to the virtual-APIC page. Due to this erratum, a VTPR write access during event delivery may cause an APIC-access VM exit with no data being written to the virtual-APIC page.

Implication: VTPR accesses are accesses to offset 80H on the APIC-access page. VTPR write accesses can occur during event delivery when pushing data on the stack. Because event delivery performs multiple stack pushes, an event delivery that includes a VTPR write access will also include at least one other write to the APIC-access page. That other write will cause an APIC-access VM exit. Thus, even in the presence of this erratum, any event delivery that includes a VTPR write access will cause an APIC-access VM exit. The only difference with respect to correct behavior will be with regard to page offset saved in the exit qualification by the APIC-access VM exit. A VMM should be able to emulate the event delivery correctly even with the incorrect offset.

Workaround: The VMM should emulate any event delivery that causes an APIC-access VM exit in the same way regardless of the offset saved in the exit qualification.

Status: For the steppings affected, see the Summary Tables of Changes.

AH110. BIST Failure after Reset

Problem: The processor may show an erroneous BIST (built-in self test) result in bit [17] of EAX register when coming out of reset.

Implication: When this erratum occurs, an erroneous BIST failure will be reported in EAX bit [17]. This failure can be ignored since it is not accurate.

Workaround: It is possible for BIOS to workaround this erratum by masking off bit [17] of the EAX register after coming out of reset.

Status: For the steppings affected, see the Summary Tables of Changes.

AH111. Performance Monitoring Event FP_MMX_TRANS_TO_MMX May Not Count Some Transitions

Problem: Performance Monitor Event FP_MMX_TRANS_TO_MMX (Event CCH, Umask 01H) counts transitions from x87 Floating Point (FP) to MMX™ instructions. Due to this erratum, if only a small number of MMX instructions (including EMMS) are executed immediately after the last FP instruction, a FP to MMX transition may not be counted.

Implication: The count value for Performance Monitoring Event FP_MMX_TRANS_TO_MMX may be lower than expected. The degree of undercounting is dependent on the occurrences of the erratum condition while the counter is active. Intel has not observed this erratum with any commercially available software.

Workaround: None Identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH112. Instruction Fetch May Cause a Livelock During Snoops of the L1 Data Cache

Problem: A livelock may be observed in rare conditions when instruction fetch causes multiple level one data cache snoops.

Implication: Due to this erratum, a livelock may occur. Intel has not observed this erratum with any commercially available software.

Workaround: It is possible for BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Tables of Changes.

AH113 Use of Memory Aliasing with Inconsistent Memory Type may Cause a System Hang or a Machine Check Exception

Problem: Software that implements memory aliasing by having more than one linear addresses mapped to the same physical page with different cache types may cause the system to hang or to report a machine check exception (MCE). This would occur if one of the addresses is non-cacheable and used in a code segment and the other is a cacheable address. If the cacheable address finds its way into the instruction cache, and the non-cacheable address is fetched in the IFU, the processor may invalidate the non-cacheable address from the fetch unit. Any micro-architectural event that causes instruction restart will be expecting this instruction to still be in the fetch unit and lack of it will cause a system hang or an MCE.

Implication: This erratum has not been observed with commercially available software.

Workaround: Although it is possible to have a single physical page mapped by two different linear addresses with different memory types, Intel has strongly discouraged this practice as it may lead to undefined results. Software that needs to implement memory aliasing should manage the memory type consistency

Status: For the steppings affected, see the Summary Tables of Changes.

AH114 A WB Store Following a REP STOS/MOVS or FXSAVE May Lead to Memory-Ordering Violations

Problem: Under certain conditions, as described in the Software Developers Manual section "Out-of-Order Stores For String Operations in Pentium 4, Intel Xeon, and P6 Family Processors", the processor may perform REP MOVS or REP STOS as write combining stores (referred to as “fast strings”) for optimal performance. FXSAVE may also be internally implemented using write combining stores. Due to this erratum, stores of a WB (write back) memory type to a cache line previously written by a preceding fast string/FXSAVE instruction may be observed before string/FXSAVE stores.

Implication: A write-back store may be observed before a previous string or FXSAVE related store. Intel has not observed this erratum with any commercially available software.

Workaround: Software desiring strict ordering of string/FXSAVE operations relative to subsequent write-back stores should add an MFENCE or SFENCE instruction between the string/FXSAVE operation and following store-order sensitive code such as that used for synchronization.

Status: For the steppings affected, see the Summary Tables of Changes.

AH115 VM Exit with Exit Reason "TPR Below Threshold" Can Cause the Blocking by MOV/POP SS and Blocking by STI Bits to Be Cleared in the Guest Interruptibility-State Field

Problem: As specified in Section, "VM Exits Induced by the TPR Shadow", in the *Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3B*, a VM exit occurs immediately after any VM entry performed with the "use TPR shadow", "activate secondary controls", and "virtualize APIC accesses" VM-execution controls all set to 1 and with the value of the TPR shadow (bits 7:4 in byte 80H of the virtual-APIC page) less than the TPR-threshold VM-execution control field. Due to this erratum, such a VM exit will clear bit 0 (blocking by STI) and bit 1 (blocking by MOV/POP SS) of the interruptibility-state field of the guest-state area of the VMCS (bit 0 - blocking by STI and bit 1 - blocking by MOV/POP SS should be left unmodified).

Implication: Since the STI, MOV SS, and POP SS instructions cannot modify the TPR shadow, bits 1:0 of the interruptibility-state field will usually be zero before any VM entry meeting the preconditions of this erratum; behavior is correct in this case. However, if VMM software raises the value of the TPR-threshold VM-execution control field above that of the TPR shadow while either of those bits is 1, incorrect behavior may result. This may lead to VMM software prematurely injecting an interrupt into a guest. Intel has not observed this erratum with any commercially available software.

Workaround: VMM software raising the value of the TPR-threshold VM-execution control field should compare it to the TPR shadow. If the threshold value is higher, software should not perform a VM entry; instead, it could perform the actions that it would normally take in response to a VM exit with exit reason "TPR below threshold".

Status: For the steppings affected, see the Summary Tables of Changes.

AH116 Using Memory Type Aliasing with Cacheable and WC Memory Types May Lead to Memory Ordering Violations

Problem: Memory type aliasing occurs when a single physical page is mapped to two or more different linear addresses, each with different memory types. Memory type aliasing with a cacheable memory type and WC (write combining) may cause the processor to perform incorrect operations leading to memory ordering violations for WC operations.

Implication: Software that uses aliasing between cacheable and WC memory types may observe memory ordering errors within WC memory operations. Intel has not observed this erratum with any commercially available software.

Workaround: None identified. Intel does not support the use of cacheable and WC memory type aliasing, and WC operations are defined as weakly ordered.

Status: For the steppings affected, see the Summary Tables of Changes.

AH117 RSM Instruction Execution under Certain Conditions May Cause Processor Hang or Unexpected Instruction Execution Results

Problem: RSM instruction execution, under certain conditions triggered by a complex sequence of internal processor micro-architectural events, may lead to processor hang, or unexpected instruction execution results.

Implication: In the above sequence, the processor may live lock or hang, or RSM instruction may restart the interrupted processor context through a nondeterministic EIP offset in the code segment, resulting in unexpected instruction execution, unexpected exceptions or system hang. Intel has not observed this erratum with any commercially available software.

Workaround: It is possible for the BIOS to contain a workaround for this erratum. Please contact your Intel sales representative for availability.

Status: For the steppings affected, see the Summary Tables of Changes.

AH118 NMIs May Not Be Blocked by a VM-Entry Failure

Problem: The *Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3B: System Programming Guide, Part 2* specifies that, following a VM-entry failure during or after loading guest state, "the state of blocking by NMI is what it was before VM entry." If non-maskable interrupts (NMIs) are blocked and the "virtual NMIs" VM-execution control set to 1, this erratum may result in NMIs not being blocked after a VM-entry failure during or after loading guest state.

Implication: VM-entry failures that cause NMIs to become unblocked may cause the processor to deliver an NMI to software that is not prepared for it.

Workaround: VMM software should configure the virtual-machine control structure (VMCS) so that VM-entry failures do not occur.

Status: For the steppings affected, see the Summary Tables of Changes.

AH119 Benign Exception after a Double Fault May Not Cause a Triple Fault Shutdown

Problem: According to the *Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3A, Exception and Interrupt Reference*, if another exception occurs while attempting to call the double-fault handler, the processor enters shutdown mode. However due to this erratum, only Contributory Exceptions and Page Faults will cause a triple fault shutdown, whereas a benign exception may not.

Implication: If a benign exception occurs while attempting to call the double-fault handler, the processor may hang or may handle the benign exception. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

AH120 IA32_MC1_STATUS MSR Bit[60] Does Not Reflect Machine Check Error Reporting Enable Correctly

Problem: IA32_MC1_STATUS MSR (405H) bit[60] (EN- Error Enabled) is supposed to indicate whether the enable bit in the IA32_MC1_CTL MSR (404H) was set at the time of the last update to the IA32_MC1_STATUS MSR. Due to this erratum, IA32_MC1_STATUS MSR bit[60] instead reports the current value of the IA32_MC1_CTL MSR enable bit.

Implication: IA32_MC1_STATUS MSR bit [60] may not reflect the correct state of the enable bit in the IA32_MC1_CTL MSR at the time of the last update.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

AH121 Corruption of CS Segment Register During RSM While Transitioning From Real Mode to Protected Mode

Problem: During the transition from real mode to protected mode, if an SMI (System Management Interrupt) occurs between the MOV to CR0 that sets PE (Protection Enable, bit 0) and the first far JMP, the subsequent RSM (Resume from System Management Mode) may cause the lower two bits of CS segment register to be corrupted.

Implication: The corruption of the bottom two bits of the CS segment register will have no impact unless software explicitly examines the CS segment register between enabling protected mode and the first far JMP. *Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3A: System Programming Guide, Part 1*, in the section titled "Switching to Protected Mode" recommends the far JMP immediately follows the write to CR0 to enable protected mode. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the Summary Table of Changes.

AH122 FP Data Operand Pointer May Be Incorrectly Calculated After an FP Access Which Wraps a 4-Gbyte Boundary in Code That Uses 32-Bit Address Size in 64-bit Mode

Problem: The FP (Floating Point) Data Operand Pointer is the effective address of the operand associated with the last non-control FP instruction executed by the processor. If an 80-bit FP access (load or store) uses a 32-bit address size in 64-bit mode and the memory access wraps a 4-Gbyte boundary and the FP environment is subsequently saved, the value contained in the FP Data Operand Pointer may be incorrect.

Implication: Due to this erratum, the FP Data Operand Pointer may be incorrect. Wrapping an 80-bit FP load around a 4-Gbyte boundary in this way is not a normal programming practice. Intel has not observed this erratum with any commercially available software.

Workaround: If the FP Data Operand Pointer is used in a 64-bit operating system which may run code accessing 32-bit addresses, care must be taken to ensure that no 80-bit FP accesses are wrapped around a 4-Gbyte boundary.

Status: For the steppings affected, see the Summary Table of Changes.

Erratum Affecting Only Intel® Core™2 Duo Mobile Processors on Mobile Intel® 965 Express Chipset Family

AH1P. VM Exit Due to Virtual APIC-Access May Clear RF

Problem: RF (Resume Flag), bit 16 of the EFLAGS/RFLAGS register, is used to restart instruction execution without getting an instruction breakpoint on the instruction following a debug breakpoint exception. Due to this erratum, in a system supporting Intel® Virtualization Technology, when a VM Exit occurs due to Virtual APIC-Access (Advanced Programmable Interrupt Controller-Access) the EFLAGS/RFLAGS saved in the VMCS (Virtual-Machine Control Structure) may contain an RF value of 0.

Implication: When this erratum occurs, following a VM Exit due to a Virtual APIC-access, the processor may unintentionally break on the subsequent instruction after VM entry.

Workaround: None identified.

Status: For the steppings affected, see the Summary Tables of Changes.

AH2P. VMCALL Failure Due to Corrupt MSEG Location May Cause VM Exit to Load the Machine State Incorrectly

Problem: In systems supporting Intel Virtualization Technology, if a VMCALL failure occurs due to a corrupt Monitor Segment (MSEG), subsequent VM Exits may load machine state incorrectly.

Implication: Occurrence of this erratum may result in a VMX abort.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the Summary Tables of Changes.

AH3P. Fixed Function Performance Counters MSR_PERF_FIXED_CTR1(30AH) and MSR_PERF_FIXED_CTR2(30BH) are Note Cleared When the Processor Is Reset

Problem: The Fixed Function Performance Counters that count the number of core cycles and reference cycles when the core is not in a halt state are not cleared when the processor is reset.

Implication: The MSR_PERF_FIXED_CTR1 and MSR_PERF_FIXED_CTR2 counters may contain unexpected values after reset.

Workaround: BIOS can workaround this erratum by clearing the counters at processor initialization time.

Status: For the steppings affected, see the Summary Tables of Changes.

AH4P Multi-Core Processors Configured for Single Core Operation May Not Be Able to Enter Intel® Enhanced Deeper Sleep

Problem: BIOS may contain the option to disable CMP (Core Multiple Processing). Disabling CMP configures a processor for single core operation. Due to this erratum, a multi-core processor operating with CMP disabled may not be able to enter Intel® Enhanced Deeper Sleep if a SIPI (Start-up Inter-Processor Interrupt) is sent to the disabled processor.

Implication: When this erratum occurs, the processor may not be able to enter the Intel® Enhanced Deeper Sleep and therefore may consume more power than expected. Intel has not observed this erratum with any commercially available system or software.

Workaround: None Identified

Status: For the affected steppings, see the Summary Tables of Changes

AH5P VTPR Access May Lead to System Hang

Problem: The logical processor may hang if an instruction performs a VTPR access and the next instruction to be executed is located on a different code page.

Implication: Software running VMX non-root operation may cause a logical processor to hang if the virtual-machine monitor (VMM) sets both the “use TPR shadow” and “virtualize APIC accesses” VM-execution controls.

Workaround: It is possible for the BIOS to contain a workaround for this erratum

Status: For the affected steppings, see the Summary Tables of Changes

AH6P Activation of Intel® Adaptive Thermal Monitor While Intel® Dynamic Front Side Bus Frequency Switching Is Active May Lead to an Incorrect Operating Point Frequency

Problem: Intel Adaptive Thermal Monitor has the ability to use multiple frequency/voltage operating points to cool the processor while maintaining a high level of performance. If Intel Dynamic Front Side Bus Frequency Switching is active, activation of the Intel Adaptive Thermal Monitor may transition the processor to the correct operating point voltage, but not frequency.
This may occur if:

1.) The software/OS requests to go to a higher performance Enhanced Intel SpeedStep® Technology operating point during the thermal monitor activation period.

2.) An entry into C4 state or Intel Enhanced Deeper Sleep interrupts the transition between the Intel Dynamic Front Side Bus Frequency Switching frequency and the targeted thermal monitor operating point frequency.

Implication: If this erratum occurs, the Intel Dynamic Front Side Bus Frequency Switching operating point frequency will be observed along with the Intel Adaptive Thermal Monitor operating point voltage. The performance state status register (IA32_PERF_STS) will reflect this intermediate performance state. There is no functional impact; the eventual voltage/frequency selection is a valid operating point. De-activation of the Intel Adaptive Thermal Monitor will result in the processor transitioning to the expected Enhanced Intel SpeedStep Technology operating point.

Workaround: None Identified

Status: For the affected steppings, see the Summary Tables of Changes

Specification Changes

AP1: The following specification change is incorporated in the *Intel® Core™2 Duo Processor and Intel® Core™2 Extreme Processor for Platforms based on Intel® 965 Express Chipset Family – Electrical, Mechanical, and Thermal Specification (EMTS)* – Rev. 2.7, Table 27. (Doc #355617):

Table 23. CMOS/Open Drain Signal and Power Up AC Specifications

T# Parameter	Min	Typ	Max	Unit	Figure	Notes[1,2,3]
T35: CMOS asynchronous input pulse width, except PWRGOOD	30			ns		6
T36: PWRGOOD to RESET# de-assertion time	1		10	ms	18	
T37: BCLK stable to PWRGOOD assertion	10			BCLKs	18	4
T38: PROCHOT# pulse width	500			µs	20	5, 12
T39: THERMTRIP# assertion to V_{CC} turn off			500	ms	23	13
T40: FERR# Valid Delay from STPCLK# deassertion	0		5	BCLKs	22	
T41: V_{CC} stable to PWRGOOD assertion	0.05	20	500	ms	18	9, 11, 14, 15
T42: PWRGOOD rise time			20	ns		7
T43: V_{CC_BOOT} stable to VID/BSEL valid			10	µs	18	8, 10, 11
T44: VID/BSEL valid to V_{CC} stable			500	ms	18	11
T48: V_{CCP} stable to VID/BSEL valid			10	µs	18	8, 11
T49: V_{CCA} stable to PWRGOOD assertion	1			ms	18	11

NOTES:

1. Unless otherwise noted, all specifications in this table apply to all processor frequencies.
2. All CMOS signal timings are referenced at VCCP/2.
3. These signals may be driven asynchronously.
4. Refer to the PWRGOOD definition for more details regarding the behavior of this signal.
5. Length of assertion for PROCHOT# does not equal thermal control circuit (TCC) activation time. Time is required after the assertion or deassertion of PROCHOT# for the processor to enable or disable the TCC. Additionally, time is allocated after the assertion or deassertion of PROCHOT# for the processor to complete current instruction execution. This specification applies to PROCHOT# as both an input (if enabled) and an output.
6. Equivalent to 2 PCI clocks duration.
7. Measured between (0.3*VCCP) and (0.7*VCCP).
8. This specification requires that the VID and BSEL signals be sampled no earlier than 10 µs after V_{CC} (at V_{CC_BOOT} voltage) and V_{CCP} are stable.
9. STPCLK#, SLP#, DPSLP# must be inactive and BCLK must toggle.
10. This signal defines the latest time between V_{CCA}, V_{CCP}, and V_{CC_CORE} ramp to stable VID valid.
11. Parameter must be measured after the last of Vcc, V_{CCA} and V_{CCP} are stable and there is no ringing around the measured voltage.
12. It may take up to 250 µs to identify the assertion of the external PROCHOT# signal and additional 250 µs to reach the low power state after identifying its assertion.
13. THERMTRIP# functionality not guaranteed if PWRGOOD is not asserted.
14. Customer platform must handle thermal control this period and should not use the THERMTRIP# functionality of the processor.
15. For Penryn processor, V_{CC_BOOT} stable to PWRGOOD assertion should not exceed 510ms. Please refer to the Penryn Processor in Santa Rosa Electrical, Mechanical and Thermal Specification for more details.

Specification Clarifications

AH1. Removed

AH2. Removed

AH3. Clarification of TRANSLATION LOOKASIDE BUFFERS (TLBS) Invalidation

Section 10.9 INVALIDATING THE TRANSLATION LOOKASIDE BUFFERS (TLBS) of the *Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3A: System Programming Guide* will be modified to include the presence of page table structure caches, such as the page directory cache, which Intel processors implement. This information is needed to aid operating systems in managing page table structure invalidations properly.

Intel will update the *Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3A: System Programming Guide* in the coming months. Until that time, an application note, TLBs, Paging-Structure Caches, and Their Invalidation (http://www.intel.com/products/processor/manuals/index.htm), is available which provides more information on the paging structure caches and TLB invalidation.

In rare instances, improper TLB invalidation may result in unpredictable system behavior, such as system hangs or incorrect data. Developers of operating systems should take this documentation into account when designing TLB invalidation algorithms. For the processors affected, Intel has provided a recommended update to system and BIOS vendors to incorporate into their BIOS to resolve this issue.

§

Documentation Changes

Note: Documentation changes for *Intel® 64 and IA-32 Architectures Software Developer's Manual* volumes 1, 2A, 2B, 3A, and 3B will be posted in a separate document *Intel® 64 and IA-32 Architectures Software Developer's Manual* Documentation Changes. Follow the link below to become familiar with this file.
http://www.intel.com/design/processor/specupdt/252046.htm

§